1330

WYLE
LABORATORIES SCIENTIFIC SERVICES & SYSTEMS GROUP

WYLE LABORATORES - RESEARCH STAFF TECHNICAL REPORT 66338-01

A DATA BASE AND ANALYSIS PROGRAM FOR SHUTTLE MAIN ENGINE DYNAMIC PRESSURE MEASUREMENTS

(NASA-CR-178808) A DATA BASE AND ANALYSIS PROGRAM FOR SHUTTLE MAIN ENGINE DYNAMIC PRESSURE MEASUREMENTS Final Report (Wyle Labs., Inc.) 192 p HC A09/MF A01 CSCL 21H
N86-24750
Unclas
G3/20 43124

research REPORT

WYLE LABORATORES - RESEARCH STAFF
TECHNICAL REPORT 66338-01

A DATA BASE AND ANALYSIS PROGRAM FOR SHUTTLE MAIN ENGINE DYNAMIC PRESSURE MEASUREMENTS

BY

THOMAS COFFIN

A final report of
work performed under contract NAS8-34343

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER
MARSHALL SPACE FLIGHT CENTER, ALABAMA 35812

JANUARY 1986

FOREWORD

This report describes a dynamic pressure data base and data base management system developed to characterize the Space Shuttle Main Engine (SSME) dynamic pressure environment. The data base represents dynamic pressure measurements obtained during single engine hot firing tests of the SSME. Software is provided to permit statistical evaluation of selected measurements under specified operating conditions. An interpolation scheme is also included to estimate spectral trends with SSME power level. This report was prepared by Wyle Laboratories Scientific Services and Systems Group for the National Aeronautics and Space Administration, George C. Marshall Space Flight Center. The work was performed under NASA contract NAS 8-34343, entitled "Flow Dynamic Environments in High Performance Rocket Engines."

The author wishes to acknowledge the contribution to this study by Dr. G. Meares, Chief Architect of the data base management software described herein. Messrs. B. Dobbs and D. Duck contributed long hours over a digitizer pad to accomplish data base input. The singular value decomposition software for spectrum interpolation was developed by Dr. J. Jong. Mr. T. Nesman, MSFC technical contract monitor, provided continuing support through informal project reviews and served as a focal point for definition of SSME data requirements.

ABSTRACT

~~This report describes~~ a dynamic pressure data base/management system for measurements obtained from space shuttle main engine (SSME) hot firing tests is described. ~~at the National Space Technology Laboratories and the Santa Susana test facility operated by Rocketdyne.~~ The data were provided ~~by NASA/MSFC~~ in terms of engine power level and rms ~~r.m.s.~~ pressure time histories, and power spectra of the dynamic pressure measurements at selected times during each test. Test measurements and engine locations are defined along with a discussion of data acquisition and reduction procedures. A description of the data base management/analysis system is provided and subroutines developed for obtaining selected measurement means, variances, ranges and other statistics of interest are discussed. A summary of pressure spectra obtained at SSME rated power level is provided for reference. Application of the singular value decomposition technique to spectrum interpolation is discussed and isoplots of interpolated spectra are presented to indicate measurement trends with engine power level. Program listings of the data base management and spectrum interpolation software are given. Appendices are included to document all data base measurements.

1330

ABSTRACT

This report describes a dynamic pressure data base/management system for measurements obtained from space shuttle main engine (SSME) hot firing tests at the National Space Technology Laboratories and the Santa Susana test facility operated by Rocketdyne. The data were provided by NASA/MSFC in terms of engine power level and r.m.s. pressure time histories, and power spectra of the dynamic pressure measurements at selected times during each test. Test measurements and engine locations are defined along with a discussion of data acquisition and reduction procedures. A description of the data base management/analysis system is provided and subroutines developed for obtaining selected measurement means, variances, ranges and other statistics of interest are discussed. A summary of pressure spectra obtained at SSME rated power level is provided for reference. Application of the singular value decomposition technique to spectrum interpolation is discussed and isoplots of interpolated spectra are presented to indicate measurement trends with engine power level. Program listings of the data base management and spectrum interpolation software are given. Appendices are included to document all data base measurements.

TABLE OF CONTENTS

UNDER SEPARATE COVER

FIGURES

PAGE

Tables

SECTION I
INTRODUCTORY SUMMARY

The Space Shuttle Main Engines (SSMEs) are designed to provide maximum performance within stringent constraints on size, weight, and efficiency. As a result the SSMEs are required to operate under extreme temperatures, with high fluid pressures and rotational pump speeds. Since the Space Shuttle Vehicle (SSV) is man-rated, and a key to the cost effectiveness of the SSV concept is hardware reusability, system reliability is of paramount importance.

Even under nominal operating conditions, the rocket engine systems and components are subject to significant pressure fluctuations. These pressure fluctuations emanate from pumps, turbines, valves, and in some instances highly turbulent propellant line flows at high dynamic pressures. These dynamic environments have temporal characteristics such that in some cases they appear as complex periodic functions and in other instances as wide-band random noise. Intense pressure oscillations have been a source of high cycle fatigue damage and SSME component failure. Measurement and empirical classification of the dynamic pressure environment is therefore desirable to define nominal SSME operating characteristics and for the diagnostic evaluation of system malfunction or failure.

Extensive dynamic pressure measurements have been acquired at critical engine locations in the course of the SSME development and certification test program. Under NASA contract NAS 8-34343 Wyle Laboratories has developed a computerized data base and analysis program to characterize the SSME dynamic pressure environment under a wide range of engine operating conditions. This report summarizes the results of this investigation. (An interim review of the

program is given in Appendix A.) The objective of this study was threefold:

1. Generate a data base of dynamic pressure spectra obtained from SSME hot firing tests measurements.

2. Develop data base management software to sort, recall and perform statistical analysis of selected measurements, to characterize the dynamic pressure environment under specified SSME operating conditions.

3. Develop and apply analytical techniques to normalize the data base and predict spectral trends with SSME operating condition.

The approach employed to accomplish these objectives is discussed in the following sections.

The SSME dynamic pressure data base is discussed in Section II. An overview of the SSME development and acceptance test program is given. Dynamic pressure measurements acquired during hot firing tests were reduced by NASA/MSFC. A brief description of data reduction procedures employed by MSFC is given and the data format provided Wyle for data base generation is described. Data base measurements and associated SSME locations are defined. Tables are included which summarize the SSME tests/measurements entered into the dynamic pressure data base.

Section III describes the SSME dynamic pressure data base management system and associated analysis software. The original software was developed for implementation on a Perkin Elmer Interdata 8/32 computer system or similar equipment. Recently the program has been rewritten for application on more readily available PC/compatible systems. Data entry/edit routines are described along with subroutines for estimating statistical means, variances, ranges, etc. of selected measurements under user defined

engine operating conditions. A program listing of the data base management/analysis software is included.

A number of the dynamic pressure measurements were available for only a limited range of engine operating power levels. However, the great majority included operation at SSME rated power level (RPL). In addition, an extensive data base of related vibration measurements at RPL has been developed by MSFC. Therefore ensemble average spectra were computed for each available measurement at RPL. These results, and their method of extraction are presented in Section IV. These summary spectra should provide a valuable reference of SSME dynamic pressure environments under nominal engine operating conditions.

Scaling and interpolation of the dynamic pressure spectra is discussed in Section V. Early in the program, it was anticipated that dynamic pressure scaling would be performed through dimensional analysis based on pertinent flow/geometric parameters. Several simple models were investigated to represent phenomena such as turbulent duct flow and blade wake excitation sources. A detailed review of data base spectra indicated, however, that the pressure environment is too complex to be adequately represented by a superposition of simple sources. Dimensional arguments were therefore abandoned in favor of a more generalized interpolation technique based on the Singular Value Decomposition (SVD) of a spectral matrix representing each measurement over the range of observed test power levels. A description of the SVD technique is given in Section 5-2, along with an example of application to a typical data base measurement. The SVD technique was applied to each measurement representing an adequate sample size and available range of test power levels. In this manner a library of interpolated spectra was generated representing each measurement in 2.5% power level increments. These

isoplots illustrate the spectral trend of each measurement with varying engine power level. Finally, a software routine was developed to permit prediction of the pressure spectral density for a desired measurement at a selected engine power level.

Plots representing the complete SSME dynamic pressure data base are included as Appendices to this report under separate cover.

SECTION II

THE SSME DYNAMIC PRESSURE DATA BASE

2.1 Background

The Space Shuttle Main Engines (SSMEs) are extremely sophisticated machines, designed to provide maximum performance within stringent constraints on size, weight, and efficiency. As a result, the SSMEs are required to operate under extreme temperatures with high fluid pressures and rotational pump speeds. Additionally, development work is presently in progress to uprate SSME performance. Since the Space Shuttle Vehicle (SSV) is man-rated, and a key to the cost effectiveness of the SSV concept is hardware reusability, system reliability is of paramount importance.

The Orbiter vehicle main propulsion system consists of three SSMEs. The SSMEs are reusable, high-performance, liquid-propellant rocket engines with variable thrust. They are ignited on the ground at launch and operate in parallel, with approximately 500 seconds total firing duration. Each of the rocket engines operates at a mixture ratio (liquid oxygen/liquid hydrogen) of 6:1 and a chamber pressure of approximately 3000 psia to produce a sea-level thrust of 375,000 pounds and a vacuum thrust of 470,000 pounds. The engines are presently throttleable over a thrust range of 60 to 109 percent of the design thrust level. This provides a higher thrust level during liftoff and the initial ascent phase, and allows Orbiter acceleration to be limited to 3 g's during the final ascent phase. The engines are gimbaled ($\pm$10.5 degrees for pitch and $\pm$8.5 degrees yaw) to provide pitch, yaw, and roll control during the Orbiter boost phase.

Significant to meeting performance requirements is the use of the staged combustion power cycle coupled with high combustion chamber pressures. In the SSME-staged combustion cycle, the propellants are partially burned at high pressure and relatively low temperature in the preburners, then completely combusted at high temperature and pressure in the main chamber before expanding through the high-area-ratio nozzle. Hydrogen fuel is used to cool all combustion devices in contact with high-temperature combustion products. An electronic engine controller automatically performs checkout, start, mainstage, and engine shutdown functions. Major components of the SSME are illustrated in Figure 2-1. A more detailed view of the SSME power head is shown in Figure 2-2. This figure provides an indication of the complexity of the SSME turbomachinery.

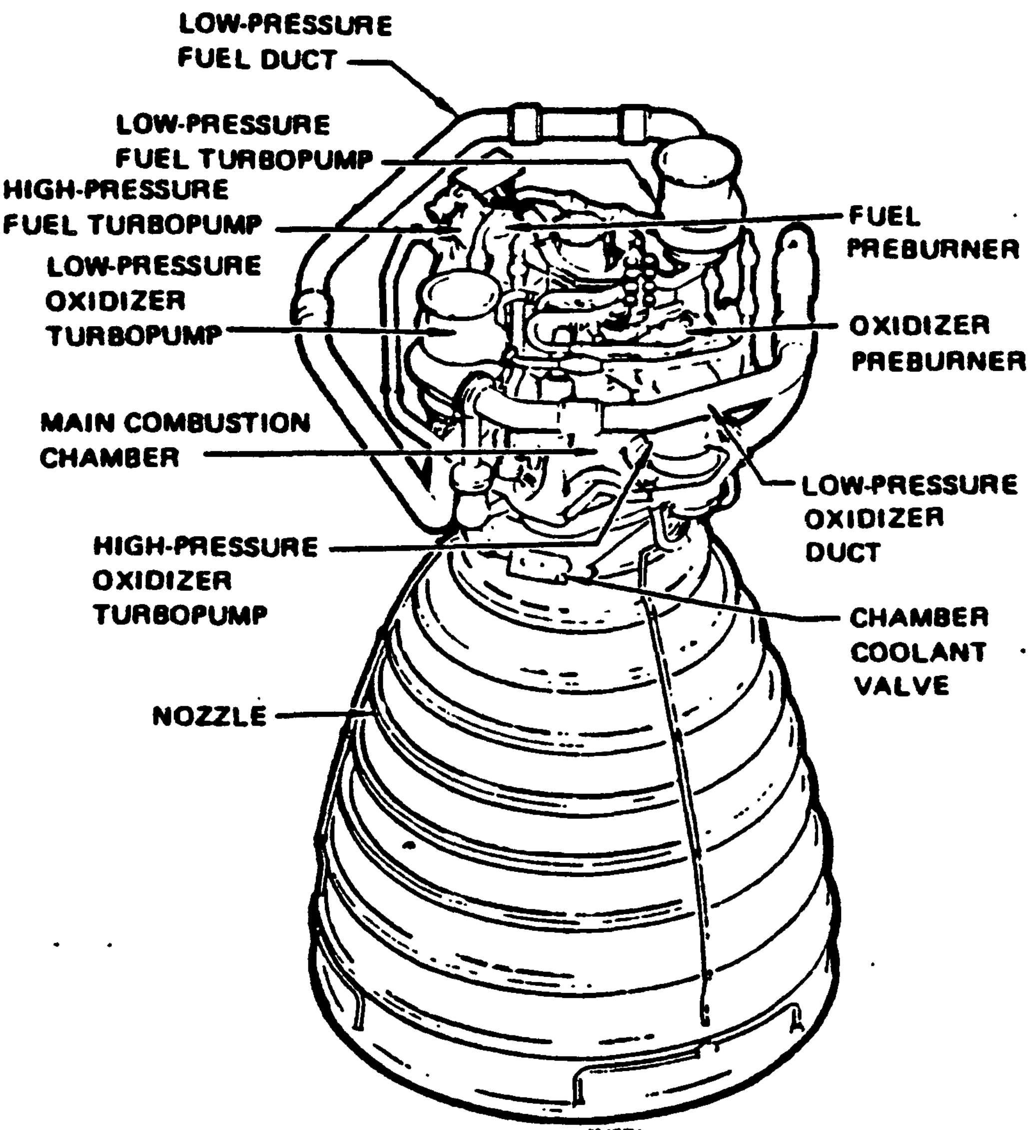


FIGURE 2-1. SPACE SHUTTLE MAIN ENGINE

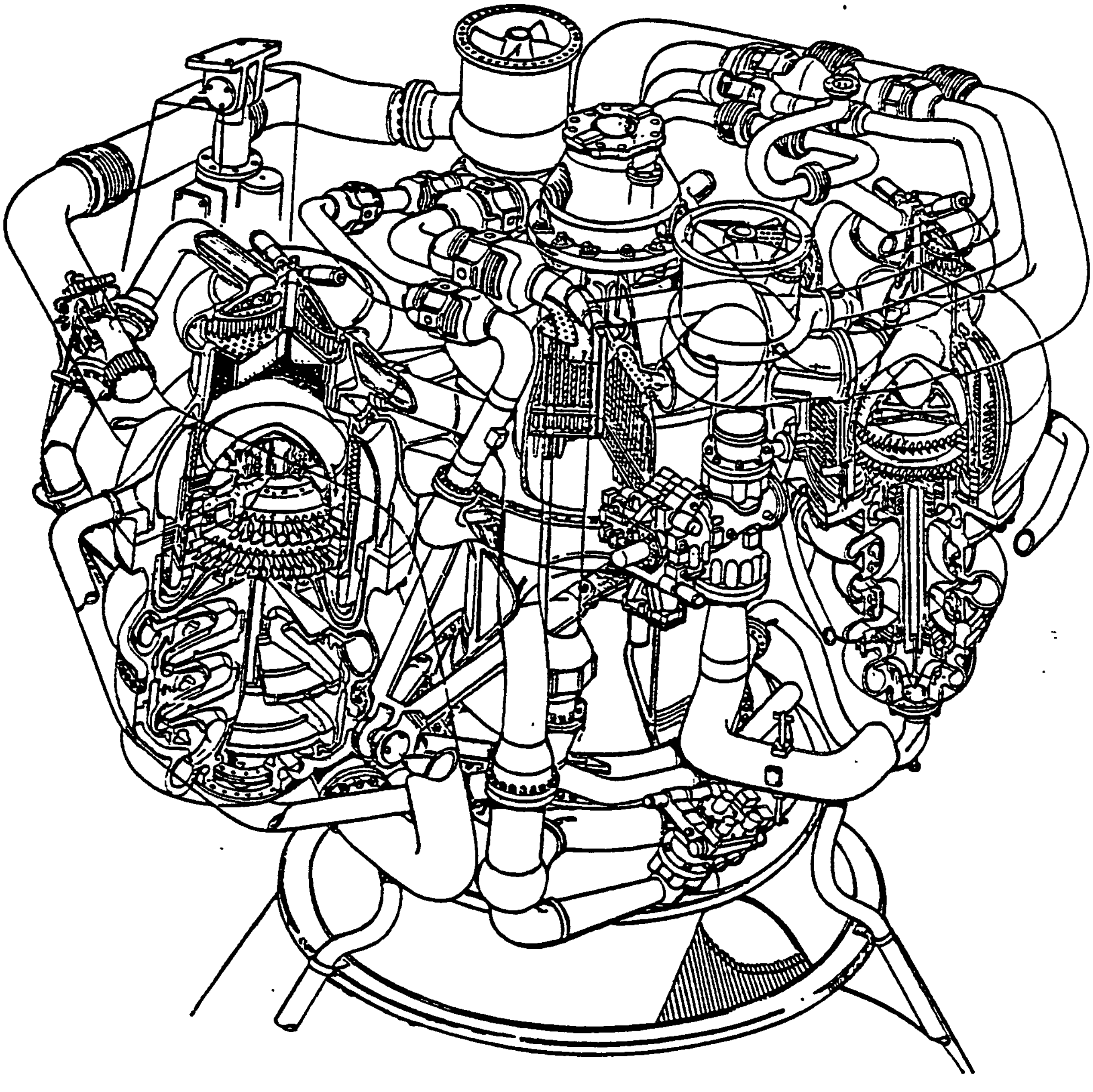


FIGURE 2-2. Space Shuttle Main Engine Power Head

2.2 SSME Development and Acceptance Testing

To validate system performance and ensure equipment reliability, the SSME and components have been and are presently undergoing extensive development and qualification tests. Testing of the engine and components is conducted at several NASA and contractor locations. Full scale engine test firings for development and flight acceptance are performed on two single-engine test stands at the National Space Technology Laboratories (NSTL), Bay St. Louis, Mississippi, and at one stand operated by Rockwell International near Santa Susana, California, with plans to include a development test stand at MSFC. In addition, main propulsion testing (MPT) is performed at NSTL on a stand designed to accommodate the Shuttle main propulsion system elements--the three-engine cluster, the ET, and the Orbiter systems.

Testing is being performed on a continuing basis. The length of a given test is dependent on specific test objectives and may run from several seconds to over 800 seconds. Tests are generally designed to satisfy multiple specific objectives, which fall into two broad categories; acceptance/certification firing of flight hardware and development testing directed towards design verification, performance and reliability improvement. Test operations are controlled by a computer called the Command and Data Simulator (CADS) which communicates with the engine, displays vital measurements for on-line observation/control and initiates pre- and post-test procedures.

Approximately 250 measurements are recorded on a given test including wide band vibration, dynamic pressure and strain at critical engine locations. Some of these measurements

are utilized on-line as emergency cut-off indicators and all are recorded on magnetic tape for subsequent analysis and evaluation. Limited SSME measurements are recorded on magnetic tape during SSV flights for evaluation with orbiter return. Typical dynamic measurements obtained during SSME operation are illustrated in Figure 2-3.

2.3 Data Base Measurements, Format and Locations

The dynamic pressure data provided by MSFC was received in three formats for defining each measurement:

1. Engine Power Level as a function of time.

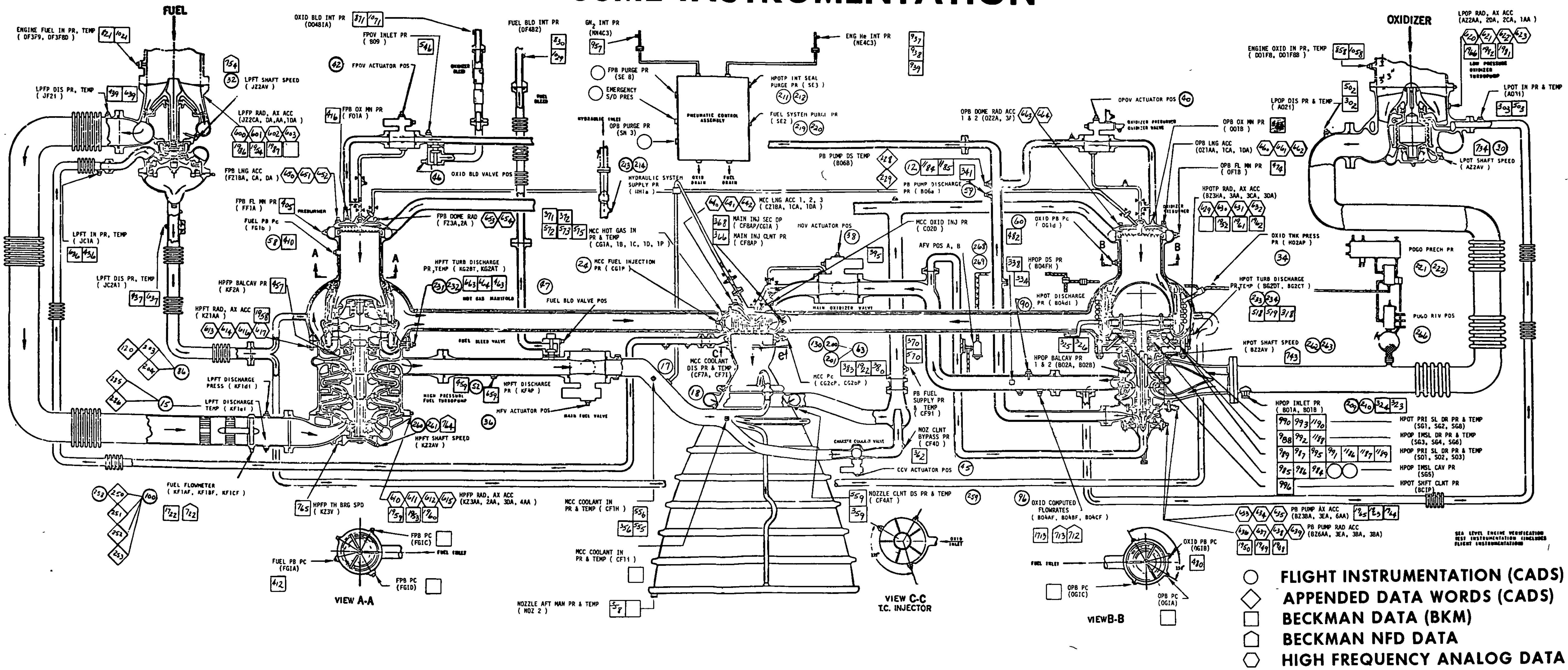
2. Root-Mean-Square pressure as a function of time (0.4 second averaging time)

3. Mean-square density spectra (PSD's) defining the frequency decomposition of each measurement at selected test conditions.

The spectral data were defined over 1-5 kHz, 1-20 kHz, and 1-40 kHz frequency bands. The great majority of data were reduced over a 1-20 kHz frequency range. The spectral data were reduced on a B&K wave analyzer yielding 400 spectral estimates over the range of analysis. This proved to be a consideration in the data base entry from continuous analog plots provided.

The dynamic pressure measurements are defined in terms of a number of acronyms for efficiency in computerized log and test description. A glossary of these abbreviations is given in Table 2-1. Table 2-2 defines the measurements and engine locations utilized in data base development. The SSME test measurement locations are illustrated in Figure 2-3. A summary of test measurements thus far entered into the data base is given in Tables 2-3, 2-4, and 2-5.

SSME INSTRUMENTATION



2/15/79

FIGURE 2-3. TYPICAL SSME MEASUREMENT LAYOUT

TABLE 2-1. GLOSSARY OF ABBREVIATIONS USED IN MEASUREMENT DESCRIPTION

LPFP	Low Pressure Fuel Pump
IN	Inlet
PR	Pressure
LPOP	Low Pressure Oxider Pump
DS	Discharge
HPOP	High Pressure Oxidizer Pump
BAL	Balance
CAV	Cavity
LOX	Liquid Oxygen
DISC	Discharge
MCC	Main Combustion Chamber
PBP	Preburner Pump (Oxidizer)
SUP	Supply
FPB	Fuel Preburner
LPOT	Low Pressure Oxidizer Turbine
TURB	Turbine
DR	Drive
HX	High Pressure
OPB	Oxidizer Preburner
HEX	High Pressure Lox
HPFP	High Pressure Fuel Pump
MAN	Manifold
INJ	Injector
PC	Chamber Pressure
DC	(Direct Current) Indicates that measurement includes mean pressure
AUX	Auxiliary
LX	Lox
HI	High Pressure
OX	Oxidizer

TABLE 2-2. DYNAMIC PRESSURE MEASUREMENT DESCRIPTIONS AND SSME LOCATION (FROM BECKMAN TEST LOG)

MEASUREMENT DESCRIPTION	SSME LOCATION
LPFP in PR	LPFP UB
LPOP DS PR	BOIB
HPOP BAL CAV PR1	B02A
HPOP DS PR	B04D1-B2
LOX IN DUCT PR2	B01B
HPOP DISC PR	BO4CI
MCC HOT GAS IN PR	CGIP
PBP DS PR	BO6A-B2
CHAMBER PR	CG2AP
LPFP DS PR	JF21
PBP FUEL SUP PR	CF91
FPB FUEL MAN PR	FF1A
LPOT TURB DR PR	AD31
LOX HX OUT PR	H02AP
OPB PC PR	OG1B
HEX OUTLET	H02AP
HPFP DS PR	KF4P
HOT GAS MAN PR3	CG1B
HPFP BAL CAV PR	KF2A
MCC FUEL INJ PR	C1P
FPB PC	FG1B-B2
MCC FUEL INJ PR DC	CG10P
OPB PC	OG1D
HPOP IN PR	BO1B
AUX LX I PR	SEQ505
HI LX I PR	BO1B
HOT GAS MAN PR2	CGID
MCC OX INJ PR	CO1H
OPB PC PR DC	OG1B

The management, analysis and interpolation of these data to characterize the SSME dynamic pressure environment is described in the following sections.

TABLE 2-3. DATA PLOT LISTING FOR TEST STAND A1

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
290	MMC HOT GAS IN PR	40,000	25
			55
			75
			195
			315
			435
			500
293	HPOP DS PR	20,000	10
			200
			400
			600
			800
	PBP DS PR		10
			200
			400
			600
			800
	OPB PC PR DC		10
			200
			400
			600
			800
294	HPOP DS PR	20,000	10
			40
			100
			300
			500
			645
			660
	PBP DS PR		10
			40
			100
			300
			500
			645
			660
	OPB PL		10
			40
			100
			300
			500
			645
			660

DATA PLOT LISTING FOR TEST STAND A1 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
294	HPOP IN PR	20,000	10
			40
			100
			300
			500
			645
			660
295	HPOP DS PR		20
			45
			100
			300
			450
			500
	PBP DS PR		20
			45
			100
			300
			450
			500
	HPOP IN PR		20
			45
			100
			300
			450
			500
	OPB PC PR DC		20
			45
			100
			300
			450
			500
297	HPOP DS PR		6
	PBP DS PR		
	OPB PC PR DC		
299	HPOP DS PR		6
	PBP DS PR		
	HPOP IN PR		
	AUX LX I PR		
	OPB PC PR DC		
300	PBP DS PR		
	HPOP IN PR		
	OPB PC PR DC		

DATA PLOT LISTING FOR TEST STAND A1 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
301	HPOP DS PR	20,000	100
			250
			400
	PBP DS PR		100
			250
			400
	OPB PC PR DC		100
			250
			400
303	HPOP DS PR		5
			30
			45
	PBP DS PR		5
			30
			45
	HPOP IN PR		5
			30
			45
	OPB PC PR DC		5
			30
			45
304	HPOP DS PR		5
			50
			75
	PBP DS PR		5
			50
			75
	LPFP DS PR		5
			50
			75
	FPB FUEL MAN PR		5
			50
			75
	HPOP IN PR		5
			50
			75
	HI LX I PR		5
			50
			75
305	HPOP DS PR		5
			25
			50
			75

DATA PLOT LISTING FOR TEST STAND A1 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
305	PBP DS PR	20,000	5
			25
			50
			75
	FPB FUEL MAN PR		5
			25
			50
			75
	HPOP IN PR		5
			25
			50
			75
	HI LX I PR		5
			25
			50
			75
306	HPOP DS PR		5
			6
			35
			70
	MCC HOT GAS IN PR	40,000	5
			20
			50
			75
	PBP DS PR	20,000	5
			35
			70
	FPB FUEL MAN PR		5
			20
			50
			75
	HPOP IN PR		5
			20
			50
			75
	HI LX I PR		5
			20
			50
			75
307	HPOP DS PR		5
			20
			45
			70

DATA PLOT LISTING FOR TEST STAND A1 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
307	MCC HOT GAS IN PR	40,000	5
			20
			45
			70
	PBP DS PR	20,000	5
			20
			45
			70
	FPB FUEL MAN PR		5
			20
			45
			70
	HPOP IN PR		5
			20
			45
			70
	HI LX I PR		5
			20
			70
309	HPOP DS PR		6
			30
	MCC HOT GAS IN PR		6
			24
	PBP DS PR		6
			30
	HPOP IN PR		6
			24
	HI LX I PR		6
			24
310	HPOP DS PR		6
			30
	MCC HOT GAS IN PR		6
			30
	PBP DS PR		6
			30
	HPOP IN PR	2,000	6
			30
	HI LX I PR	20,000	6
			30
311	HPOP DS PR		6
			30
	MCC HOT GAS IN PR		6
			30
	PBP DS PR		6
			30

DATA PLOT LISTING FOR TEST STAND A1 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
311	HPOP IN PR	20,000	6
			30
	HI LX I PR		6
			30
312	MCC HOT GAS IN PR		6
			40
	HPOP IN PR		6
			40
	HI LX I PR		6
313	MCC HOT GAS IN PR		4
			30
	HPOP IN PR		4
			30
	HI LX I PR		3
			30
314	HPOP DS PR		6
			30
	MCC HOT GAS IN PR		6
			30
		40,000	6
			30
	PBP DS PR	20,000	6
			30
	HPOP IN PR		6
			30
	HI LX I PR		6
			30
315	HPOP DS PR		4
			30
	MCC HOT GAS IN PR		30
			60
	PBP DS PR		4
			30
316	HPOP DS PR		4
			11
	MCC HOT GAS IN PR		4
			11
		40,000	4
			11
	PBP DS PR	20,000	4
			11
	OPB PC PR DC		4
			11
317	HPOP DS PR		4
			11

DATA PLOT LISTING FOR TEST STAND A1 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
317	MCC HOT GAS IN PR	20,000	4
			11
		40,000	4
			11
	PBP DS PR	20,000	4
			11
	OPB PC PR DC		4
			11
319	HPOP DS PR		6
			30
			92
	MCC HOT GAS IN PR		6
			30
			92
	PBP DS PR		6
			30
			92
	MCC IN PR	40,000	12
			60
			180
	OPB PC PR DC	20,000	6
			30
			92
321	MCC HOT GAS IN PR		6
			30
			90
			100
			270
		40,000	6
			30
			90
			100
			270
	FPB PC	20,000	6
			30
			90
			100
			270
	OPB PC PR DC		6
			30
			90
			100
			270

DATA PLOT LISTING FOR TEST STAND A1 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
322	HPOP DS PR	20,000	6
			130
			170
			190
			210
			230
			260
			280
	MCC HOT GAS IN PR		6
			130
			150
			170
			190
			210
			230
			260
			280
	PBP DS PR		6
			130
			150
			170
			190
			210
			230
			264
			284
	FPB PC		6
			130
			150
			170
			190
			210
			230
			260
			280
	OPB PC PR DC		6
			130
			150
			170
			190
			210
			230
			260
			280

DATA PLOT LISTING FOR TEST STAND A1 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
384	THRUST		
385	↓		
388	↓		
389	↓		
390	↓		
391	↓		
393	↓		
394	↓		
395	↓		
397	↓		
398	↓		
400	↓		
401	↓		
402	↓		
403	↓		
404	↓		
↓	HPOP BAL CAV PR2	20,000	20
↓	↓	↓	100
↓	↓	↓	220
↓	HPOP BAL CAV PR1	↓	20
↓	↓	↓	100
↓	↓	↓	220
↓	OPB PC PR	↓	20
↓	↓	↓	100
↓	↓	↓	220
↓	HPFP BAL CAV PR	↓	20
↓	↓	↓	100
↓	↓	↓	220
↓	FPB PC	↓	20
↓	↓	↓	100
↓	↓	↓	220
405	THRUST		
406	↓		
407	↓		
↓	HPOP BAL CAV PR2	20,000	20
↓	↓	↓	45
↓	↓	↓	75
↓	↓	↓	200
↓	↓	↓	400
↓	↓	↓	430
↓	HPOP BAL CAV PR1	↓	20
↓	↓	↓	45
↓	↓	↓	75
↓	↓	↓	200
↓	↓	↓	400
↓	↓	↓	430

DATA PLOT LISTING FOR TEST STAND A1 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
407	FPB PC	20,000	20
			45
			75
			200
			400
			430
	OPB PC PR		20
			45
			75
			200
			400
			430
	HPFP BAL CAV PR		20
			45
			75
			200
			400
			430
408	THRUST		
409	HPOP BAL CAV PR2	20,000	15
			240
			500
			525
			705
			725
	HPOP BAL CAV PR1		15
			240
			500
			525
			705
			725
	OPB PC PR		15
			240
			500
			525
			705
			725
	HPFP BAL CAV PR		15
			240
			500
			525
			705
			725

DATA PLOT LISTING FOR TEST STAND A1 (Concluded)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
409	FPB PC	20,000	7
↓	↓	↓	240
			500
			525
			705
			725
410	THRUST		
413	↓		
414			

TABLE 2-4. DATA PLOT LISTING FOR TEST STAND A2

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
214	MMC FUEL INJ PR	40,000	25
			50
			70
			150
			300
			450
			500
215			30
			50
			70
251	THRUST		
252			
253			
254			
255			
256			
257			
258			
259			
260			
261			
262			
263			
264			
265			
266			
267			
268			
269			
270			
271			
276			
277			
279			
283			
284			
303			
306			
307			
310			
311			
313			
314			

TABLE 2-5. DATA PLOT LISTING FOR TEST STAND A3

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
047	THRUST		
048			
049			
050			
052			
053			
066			
067			
068			
069			
070			
071			
072			
073			
074			
075			
076			
077			
078			
079			
087			
088			
089			
090			
091			
092			
093			
094			
095			
096			
097			
098			
099			
100			
	LPFP IN PR		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	LPFP IN PR	5,000	10
			40
			85
			95

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
100	LPFP IN PR	20,000	10
			40
			85
			95
	LPOP DS PR		10
			40
			85
			95
	HPOP BAL CAV PR2		10
			40
			85
			95
	HPOP BAL CAV PR1		10
			40
			85
			95
101	THRUST		
	LPFP IN PR		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	LPFP IN PR	5,000	5
			30
			80
			95
		20,000	5
			30
			80
			95
	LPOP DS PR		5
			30
			80
			95
	HPOP BAL CAV PR2		5
			30
			80
			95
102	THRUST		
	LPFP IN PR		
	HPOP BAL CAV PR2		
	HPOP DS PR		
	LPFP IN PR	20,000	15
			35
			60
			80

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
102	LPFP IN PR	20,000	100
			200
			280
	HPOP BAL CAV PR2		15
			35
			60
			80
			100
			150
			200
			280
	HPOP DS PR		15
			35
			60
			80
			100
			200
			280
	LOX IN DUCT PR2	5,000	15
			35
			60
			200
			280
		20,000	15
			35
			60
			80
			100
			200
			280
103	THRUST		
	LPFP IN PR		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	HPOP DISC PR		
	LPFP IN PR	5,000	5
			15
			20
			26
			30
		20,000	5
			15
			20

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
103	LPFP IN PR	20,000	26
			30
	LPOP DS PR		5
			15
			20
			26
			30
	HPOP BAL CAV PR2		5
			15
			20
			26
			30
	HPOP DISC PR		5
			15
			20
			26
			30
	MCC HOT GAS IN PR		5
			15
			20
			26
			30
		40,000	5
			15
			20
			26
			30
	MMC LONG ACCEL 1		5
104	THRUST		
	LPFP IN PR		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	HPOP DISC PR		
	MMC HOT GAS IN PR		
	LPFP IN PR	20,000	6
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP DISC PR		
	MMC HOT GAS IN PR		
		40,000	
	PBP DS PR	20,000	
105	THRUST		
	LPFP IN PR		

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
105	LPOP DS PR		
↓	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	HPOP DS PR		
	HPOP DISC PR		
	MMC HOT GAS IN PR		
	PBP DS PR		
	LPFP IN PR	20,000	6
	LPOP DS PR	↓	↓
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	HPOP DS PR		
	HPOP DISC PR		
	MMC HOT GAS IN PR	↓	
	↓	40,000	
	PBP DS PR	20,000	
↓	CHAMBER PR	↓	↓
106	THRUST		
↓	LPFP IN PR		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	HPOP DS PR		
	HPOP DISC PR		
	MCC HOT GAS IN PR		
	PBP DS PR		
	LPFP IN PR	5,000	6
		↓	11
		20,000	6
	↓	↓	11
	LPOP DS PR		6
	↓		11
	HPOP BAL CAV PR2		6
	↓		11
	HPOP BAL CAV PR1		11
	HPOP DS PR		6
	↓		11
	HPOP DISC PR		6
	↓		11
	PBP DS PR		6
↓	↓	↓	11
107	THRUST		

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
108	THRUST		
	LPFP IN PR	5,000	6
			12
		20,000	6
			12
	LPOP DS PR		6
			12
	HPOP DS PR		6
			12
	HPOP DISC PR		6
			12
	MCC HOT GAS IN PR		6
		40,000	6
			12
	PBP DS PR	20,000	6
			12
109	THRUST		
	LPFP IN PR		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	HPOP DS PR		
	HPOP DISC PR		
	MCC HOT GAS IN PR		
	PBP DS PR		
	LPFP IN PR	20,000	6
			14
	LPOP DS PR		6
			14
110	THRUST		
	LPFP DS PR		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	CHAMBER PR		
	PBP FUEL SUP PR		
	FPB FUEL MAN PR		
	LPFP DS PR	20,000	10
			30
			150
			300
			400
	HPOP BAL CAV PR2		10
			30
			150

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
110	HPOP BAL CAV PR2	20,000	300
			400
	HPOP DS PR		10
			30
			150
			300
	MCC HOT GAS IN PR		10
			30
			150
		40,000	10
			30
			150
111	THRUST		
	LPFP DS PR		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	HPOP DS PR		
	BPB DS PR		
	CHAMBER PR		
	PBP FUEL SUP PR		
	FPB FUEL MAN PR		
	LOX HX OUT PR		
	LPFP DS PR	20,000	15
			40
			50
			60
			70
			80
			90
			100
			120
			200
			280
	LPOP DS PR		15
			40
			50
			60
			70
			80
			90
			100
			120
			200
			280

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
111	HPOP BAL CAV PR2	20,000	15
			40
			50
			60
			70
			80
			90
			100
			120
			200
			280
	HPOP BAL CAV PR1		15
			40
			50
			60
			70
			80
			90
			100
			120
			200
			280
	HPOP DS PR		15
			40
			50
			60
			70
			80
			90
			100
			120
			200
			280
	PBP DS PR		15
			40
			50
			60
			70
			80
			90
			100
			120
			200
			280

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
111	PBP FUEL SUP PR	20,000	15
			40
			50
			60
			70
			80
			90
			100
			120
			200
			280
	LPOT TURB DR PR	5,000	15
			40
			80
			90
			100
			120
			200
		20,000	15
			40
			50
			60
			70
			80
			90
			100
			120
			200
			280
	LOX HX OUT PR	5,000	15
			40
			50
			60
			70
			80
			90
			100
			120
			200
			280
		20,000	40
			50
			60
			70

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
111	LOX HX OUT PR	20,000	80
			90
			100
			120
			200
			280

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
112	THRUST		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	HPOT TURB DR PR		
	LPFP DS PR	20,000	30
			150
			280
	LPOP DS PR		30
			150
			280
	HPOP BAL CAV PR2		30
			150
			280
	MCC HOT GAS IN PR		30
			150
			280
		40,000	30
			150
			280
	LPOT TURB DR PR	5,000	30
			150
			280
		20,000	30
			150
			280
	OPB PC PR		30
			150
			280
	HEX OUTLET		30
			150
			280
113	THRUST		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	HPOP DS PR		
	MCC HOT GAS IN PR		
	PBP DS PR		
	LPOP DS PR	20,000	30
			150
			280
	HPOP BAL CAV PR2	5,000	150
			280

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
113	HPOP BAL CAV PR2	20,000	30
			150
			280
	HPOP DS PR	5,000	30
			150
			280
		20,000	30
			150
			280
	MCC HOT GAS IN PR		30
			150
			280
		40,000	30
			150
	PBP DS PR	5,000	30
			150
			280
		20,000	30
			150
			280
	LPFP DS PR		30
			150
			280
	LPOT TURB DR PR	5,000	30
			150
			280
		20,000	30
			150
			280
	OPB PC PR	5,000	30
			150
			280
		20,000	30
			150
			280
	MCC LONG ACCEL 1	40,000	280
	OPB PC PR DC	5,000	30
			150
			280
		20,000	30
			150
			280
114	THRUST		
	LPOP DS PR		
	HPOP BAL CAV PR2		

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
114	HPOP BAL CAV PR1		
	HPOP DS PR		
	MCC HOT GAS IN PR		
	PBP DS PR		
	LPOT TURB DR PR		
	LPOP DS PR	20,000	30
			150
			280
	HPOP BAL CAV PR2		30
			150
			280
	HPOP DS PR		30
			150
			280
	MCC HOT GAS IN PR		30
			150
			280
	PBP DS PR		30
			150
			280
	LPFP DS PR		30
			150
			280
	LPOT TURB DR PR	5,000	30
			150
			280
		20,000	30
			150
			280
	OPB PC PR		30
			150
			280
	HPOP DS PR		30
			150
			280
115	THRUST		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	MCC HOT GAS IN PR		
	PBP DS PR		
	HPFP DS PR		
	LPOP DS PR	20,000	30
			150
			280

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
115	HPOP BAL CAV PR2	5,000	30
			150
			280
		20,000	30
			150
			280
	MCC HOT GAS IN PR		30
			150
			280
		40,000	30
			150
			280
	PBP DS PR	20,000	30
			150
			280
	LPFP DS PR		30
			150
			280
	LPOT TURB DR PR	5,000	30
			150
			280
		20,000	30
			150
			280
	OPB PC PR		30
			150
	HPFP DS PR		30
			150
			280
	HOT GAS MAN PR3		30
			150
			280
	OPB PC PR DC		30
			150
			280
116	THRUST		
	LPOP DS PR		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	HPOP DS PR		
	MCC HOT GAS IN PR		
	PBP DS PR		
	OPB PC PR		
	HPFP DS PR		
	HOT GAS MAN PR3		
	HPFP BAL CAV PR		

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
116	LPOP DS PR	5,000	30
			150
			280
		20,000	30
			150
			280
	HPOP BAL CAV PR2	5,000	30
			150
			280
		20,000	30
			150
			280
	HPOP DS PR		30
			150
			280
	MCC HOT GAS IN PR		30
			150
			280
		40,000	30
			150
			280
	PBP DS PR	5,000	30
			150
			280
		20,000	30
			150
			280
	LPFP DS PR		30
			150
			280
	LPOT TURB DR PR	5,000	30
			150
			280
		20,000	30
			150
			280
	OPB PC PR		30
			150
			280
	HPFP DS PR		30
			150
			280
	HOT GAS MAN PR3		30
			150
			280

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
116	HPFP BAL CAV PR	20,000	30
			150
			280
	OPB PC PR DC		4
			30
			150
			280
117	THRUST		
	LPOP DS PR		
	HPOP DS PR		
	MCC HOT GAS IN PR		
	PBP DS PR		
	LPFP DS PR		
	LPOT TURB DR PR		
	OPB PC PR		
	HPFP DS PR		
	HPFP BAL CAV PR		
	LPOP DS PR	5,000	30
			150
			297
		20,000	30
			150
			297
	HPOP DS PR		30
			150
			297
	MCC HOT GAS IN PR		30
			150
			297
		40,000	30
			150
			280
	PBP DS PR	20,000	30
			150
			297
	LPFP DS PR		30
			150
			297
	LPOT TURB DR PR		30
			150
			297
	OPB PC PR		30
			150
			297

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
117	HPFP DS PR	20,000	30
			150
			297
	HPFP BAL CAV PR		30
			150
			297
	OPB PC PR DC		30
			150
			297
118	THRUST		
	LPOP DS PR		
	HPOP BAL CAV PR1		
	PBP DS PR		
	LPOP DS PR	20,000	30
			150
			280
	MCC HOT GAS IN PR		30
			150
			280
		40,000	30
			150
			280
	PBP DS PR	20,000	30
			150
			280
	HPFP BAL CAV PR		30
			150
			280
	OPB PC PR DC		30
			150
			280

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
120	THRUST		
	HPOP BAL CAV PR2		
	HPOP BAL CAV PR1		
	HPOP DS PR		
	MCC HOT GAS IN PR		
	PBP DS PR		
	LPOP DS PR	20,000	30
			100
			200
	HPOP DS PR		30
			100
			200
	MCC HOT GAS IN PR		30
			100
			200
		40,000	30
			100
			200
	PBP DS PR	20,000	30
			100
			200
	HPFP BAL CAV PR		30
			100
			200
	OPB PC PR DC		30
			100
			200
121	THRUST		
	LPOP DS PR	20,000	8
			16
	HPOP DS PR		8
			16
	MCC HOT GAS IN PR		8
			16
		40,000	8
			16
	HOT GAS MAN PR2	20,000	8
			16
	HPFP BAL CAV PR		8
			16
	OPB PC PR DC		8
			16

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
123	THRUST		
124	↓		
125			
↓	MCC FUEL INJ PR	20,000	6
	↓	↓	95
			125
			155
			185
			215
			245
			265
			285
		40,000	6
		↓	10
			50
			95
			125
			155
			185
			215
			245
			265
			285
126	THRUST		
127	↓		
↓	FPB PC	20,000	6
	↓	↓	11
128	THRUST		
↓	FPB PC	20,000	6
	OPB PC PR DC	↓	↓
129	THRUST		
↓	FPB PC	20,000	6
	↓	↓	11
	MCC FUEL INJ PR	40,000	5
	↓	↓	11
	OPB PC PR DC	20,000	6
	↓	↓	11
130	THRUST		
↓	FPB PC	20,000	6
	↓	↓	11
	MCC FUEL INJ PR		6
	↓		11
		40,000	6
		↓	11

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
130	OPB PC PR DC	20,000	6
			11
131	THRUST		
	MCC FUEL INJ PR	20,000	30
			70
			100
	FPB PC		30
			70
			100
		40,000	30
			70
			100
	OPB PC PR DC	20,000	30
			70
			100
132	THRUST		
	MCC FUEL INJ PR		
	OPB PC PR DC		
	FPB PC	20,000	10
			90
			150
			260
			291
	MCC FUEL INJ PR		10
			90
			150
			260
			291
		40,000	10
			90
			150
			260
			291
	OPB PC PR DC	20,000	10
			90
			150
			260
			291
133	THRUST		
	FPB PC	20,000	6
			71
			76
			150
			280

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
133	OPB PC PR DC	20,000	6
			71
			76
			150
			280
134	THRUST		
	MCC FUEL INJ PR	20,000	4
			30
			45
			150
			280
		40,000	4
			30
			45
			150
			200
	FPB PC	20,000	4
			30
			55
			150
			280
	MCC FUEL INJ PR DC		4
			30
			45
			150
			280
		40,000	4
			30
			45
			150
			280
	OPB PC PR DC	20,000	4
			30
			45
			150
			280
135	THRUST		
	MCC FUEL INJ PR	20,000	6
			15
			45
			75
			150
		40,000	6
			15
			45

DATA PLOT LISTING FOR TEST STAND A3 (Continued)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
135	MCC FUEL INJ PR	40,000	75
↓	↓	↓	150
	MCC FUEL INJ PR DC	20,000	6
	↓	↓	15
			45
			75
			150
		40,000	6
		↓	15
			45
			75
			150
	OPB PC PR DC	20,000	6
	↓	↓	15
			45
			75
			150
136	THRUST		
↓	MCC FUEL INJ PR	20,000	4
	↓	↓	25
			50
			150
			280
		40,000	4
		↓	25
			50
			150
			280
	MCC FUEL INJ PR DC	20,000	4
	↓	↓	25
			50
			150
			280
	OPB PC PR DC		25
	↓		50
			150
			280
138	THRUST		
139	↓		
140			
141			
142			
143			
144			
145			
146			

DATA PLOT LISTING FOR TEST STAND A3 (Concluded)

Test	Channel Descriptor (Time History or PSD)	Maximum Frequency (Hz)	Time Slice (Sec)
147	THRUST		
148			
150			
151			
152			
153			
154			
155			
156			
157			
158			
161			
164			
165			
166			
167			
168			
170			
171			
172			
173			
174			
175			
177			
178			
179			
180			
181			
182			
183			
184			
185			
186			
187			
188			
190			
191			
192			
193			
194			
195			
196			
197			
198			
199			
200			

Section III

THE SSME DYNAMIC PRESSURE DATA BASE MANAGEMENT AND ANALYSIS SYSTEM

A primary objective of this study was to develop a data base and data base management system for SSME dynamic pressure measurements provided by MSFC. The data was obtained in the form of time histories (power level and r.m.s. pressure) and power spectra (PSDs). The data base management and analysis system was originally designed for use on an Interdata 8/32 computer system and other computer systems utilizing similar file name structures. The data base/management system has recently been converted for application on P.C. network systems. The data management system is designed to store test data in a manner that will provide rapid and easy access to the data by the display and analysis programs. The system can accept time history data and PSD or frequency domain data. Each time history and PSD is stored in a separate data file. The data base management system creates and names these files when the data is entered. The system also makes entries in several directory files so that the data can be quickly accessed, sorted, statistically summarized and or plotted as required by the user.

Figures 3-1 through 3-3 show sample plots of data available in the data base. Separate volumes of plots of each data file in the data base have been prepared, and are provided in appendices to this report.

This section describes the data base management system, the software that has been developed to enter data into the data base system, and the software that has been developed to analyze the data. As this software was developed, general purpose subroutines were also developed. Subroutines to

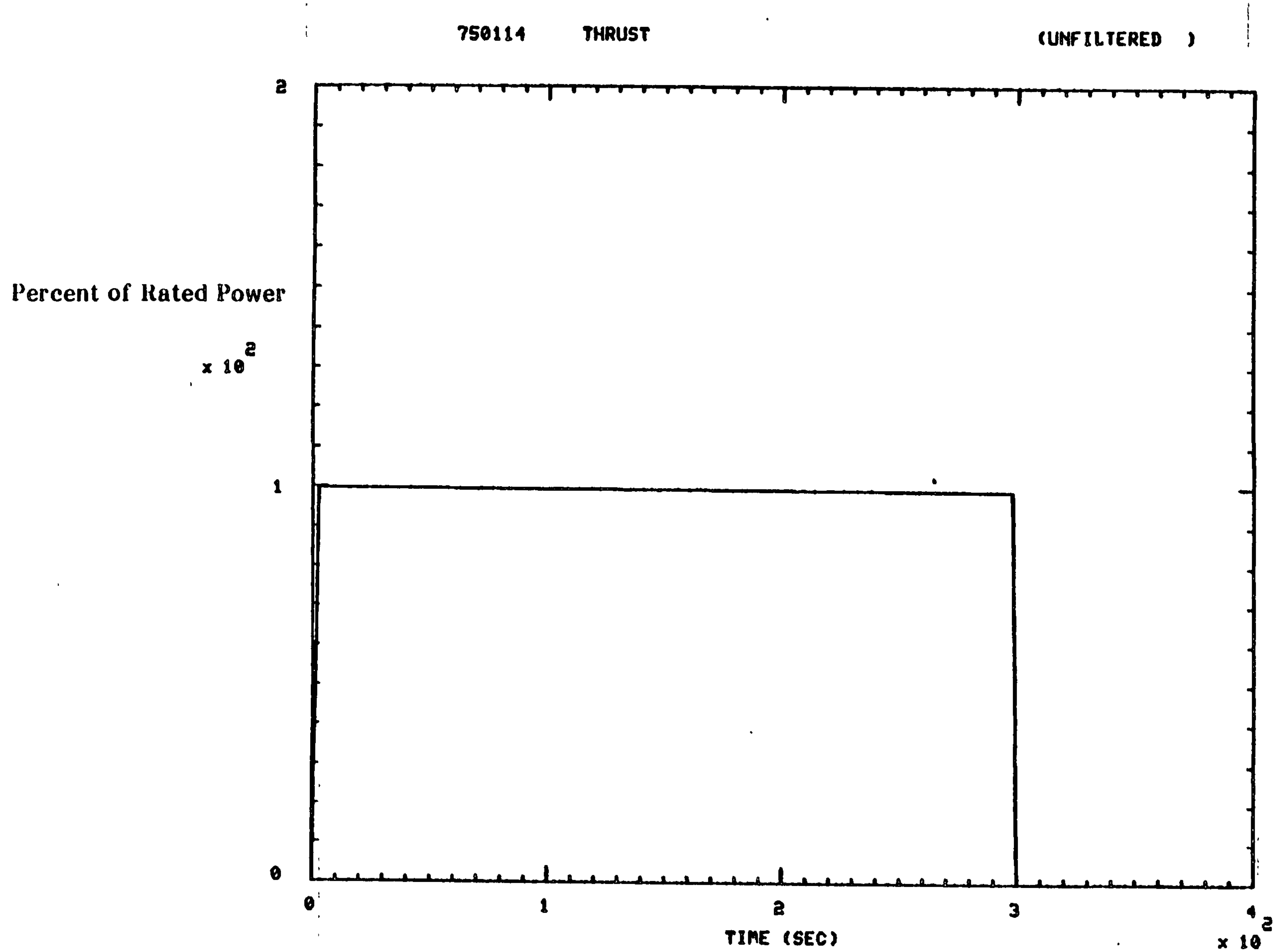


FIGURE 3-1. THRUST TIME HISTORY

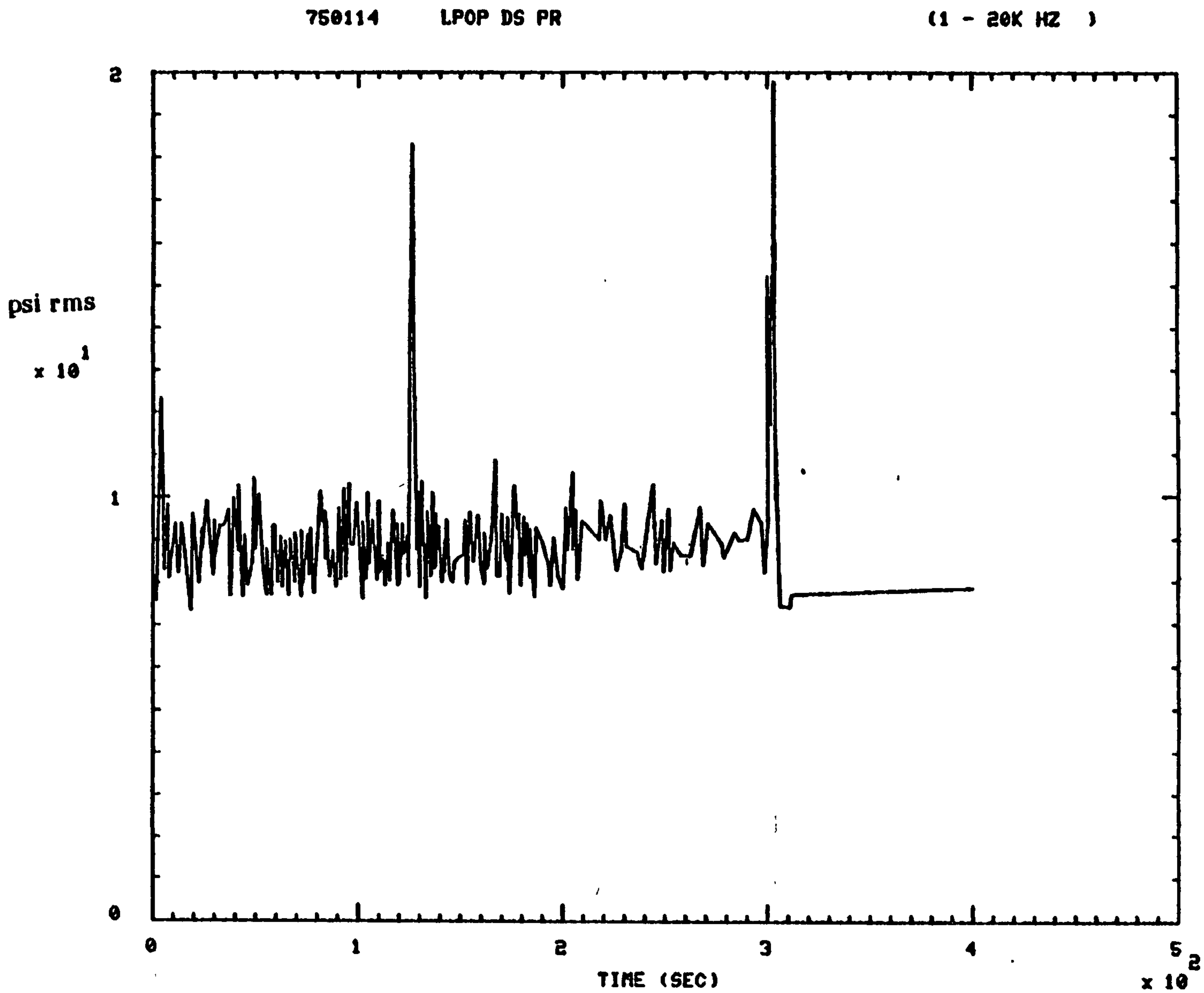


FIGURE 3-2. PRESSURE TIME HISTORY

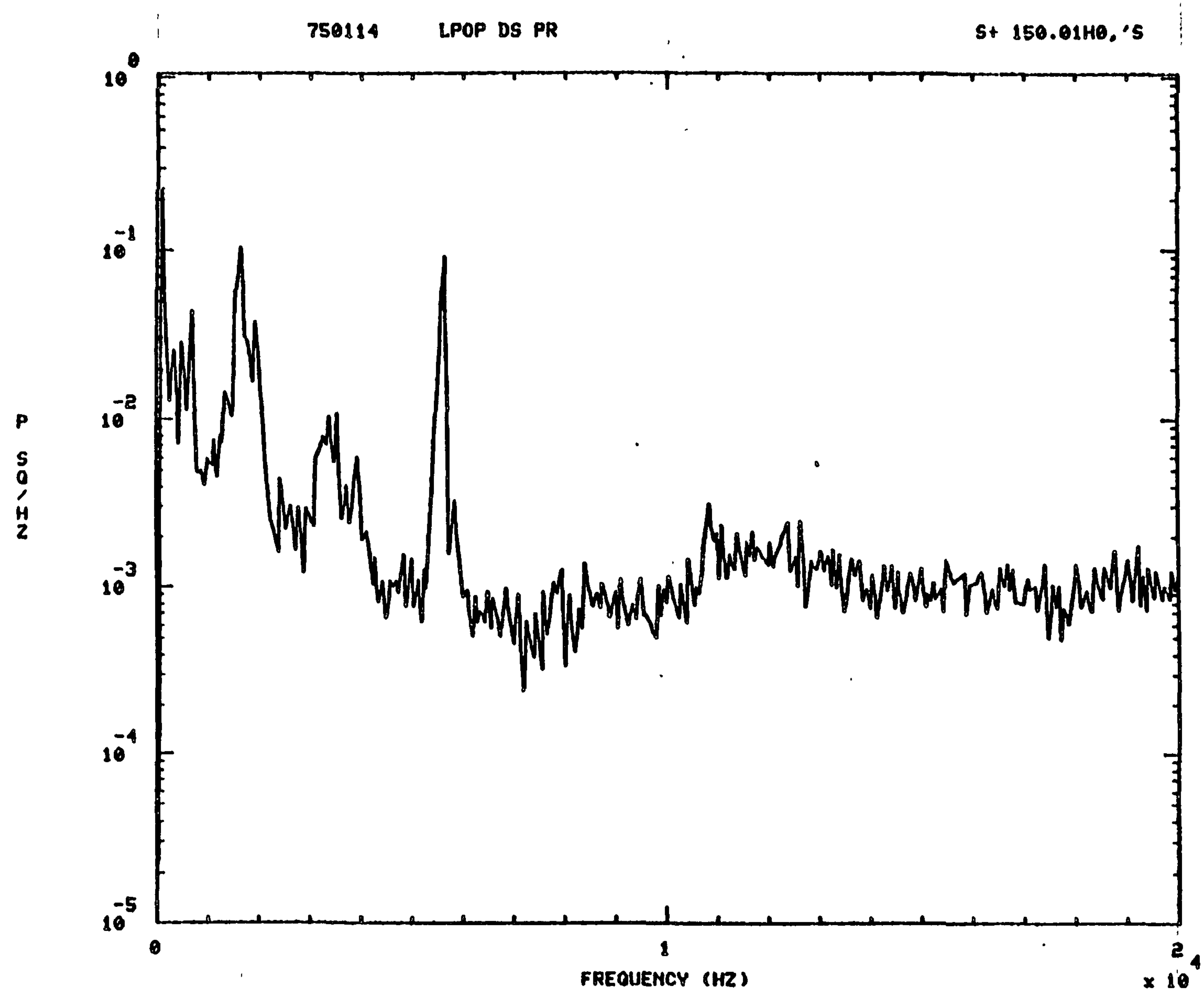


FIGURE 3-3. PRESSURE PSD

manipulate and retrieve data from the data base management system are also described.

3.1 Data Entry

Initially data was provided in the form of PSD and time history plots. A digitizer pad was used to convert the plots to digitized data. Computer programs (DIGTIME and DIGPSD) were written to interface with the digitizer pad and to enter the data into the data base management system. Both programs function interactively with the digitizer pad and its operator. A complete functional description of these programs is provided in the following section.

Over 300 time histories and 1200 PSDs have been digitized and entered into the data base management system. These plots were initially digitized by starting at the point where the trace intersected the y-axis then moving along the trace in straight-line segments, digitizing points between each segment. This resulted in data being taken at random time intervals for time histories and random frequency intervals for PSDs. The random time intervals are not considered to be a problem. For the PSDs, however, analysis efforts would be facilitated if the data was stored at uniform frequency intervals. The decision was made to convert this data to 400-line spectra to match the data reduction format in use at MSFC. Several methods were considered for making this conversion. The simplest was to interpolate between digitized points to obtain the necessary 400 points at specific frequency intervals. As shown in Figure 3-4, this method resulted in inaccuracies in the magnitudes of the peaks. Several modifications of this method were considered, but the method that proved to be the most accurate utilized the following steps:

1. Eliminate duplicated digitized points. Points that are within one-fourth of a bandwidth and within 10

percent in magnitude of each other are considered duplicates.

2. For each of the 400 bandwidths in which there is one and only one digitized point, assign that point to that bandwidth.

3. For each of the 400 bandwidths in which there are two or more digitized points, assign the lowest frequency point to the next lower bandwidth if it is unoccupied. Assign the second from the lowest frequency point within the bandwidth to that bandwidth, and if there is a third point, assign it to the next higher bandwidth if it is unoccupied. If the next lower bandwidth is occupied, then assign the lowest frequency point in the bandwidth to that bandwidth, and assign the highest frequency point in that bandwidth to the next higher bandwidth if it is unoccupied. Digitized points that cannot be assigned to a bandwidth by this method are not used.

4. Use interpolation to obtain points for bandwidths that remain unoccupied.

This conversion method was implemented in the subroutine CV400. Figure 3-5 shows a comparison of digitized PSD data to the converted 400-line spectra.

More recently, a method of entering data into the data base via magnetic tape has been developed. Time history and PSD data is transferred from a Norland analyzer to an HP 1000 computer system at MSFC. The data is then put on magnetic tape, and the tape is read by the computer system at Wyle. Software was developed to read these tapes and enter the data directly into the data base management system. This method of data transfer has proven to be more accurate and significantly more efficient than the manual digitizing

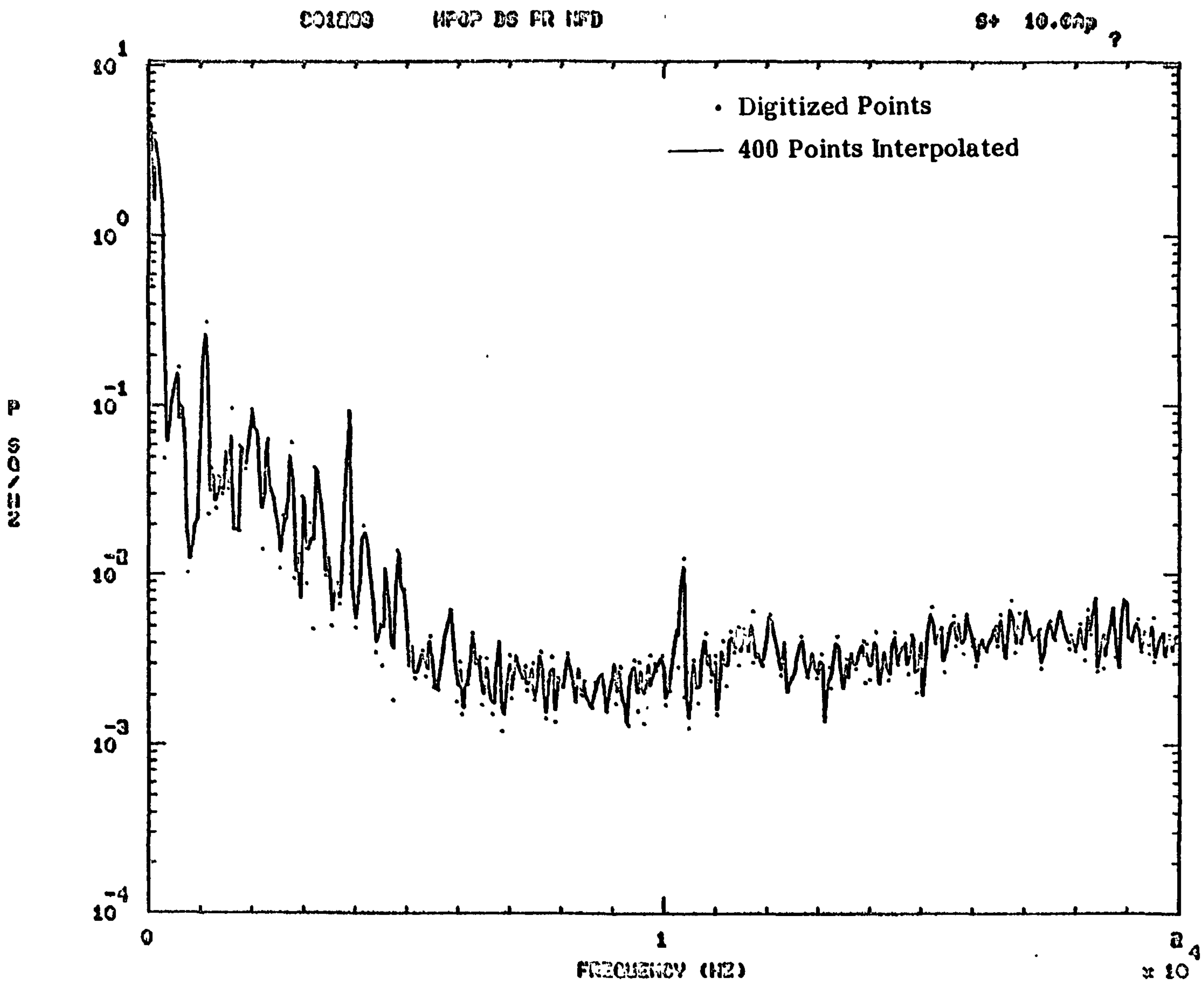


FIGURE 3-4. COMPARISON OF DIGITIZED POINTS TO 400-LINE SPECTRA OBTAINED BY INTERPOLATION

method used initially. In this mode, spectrum interpolation is obviously not required.

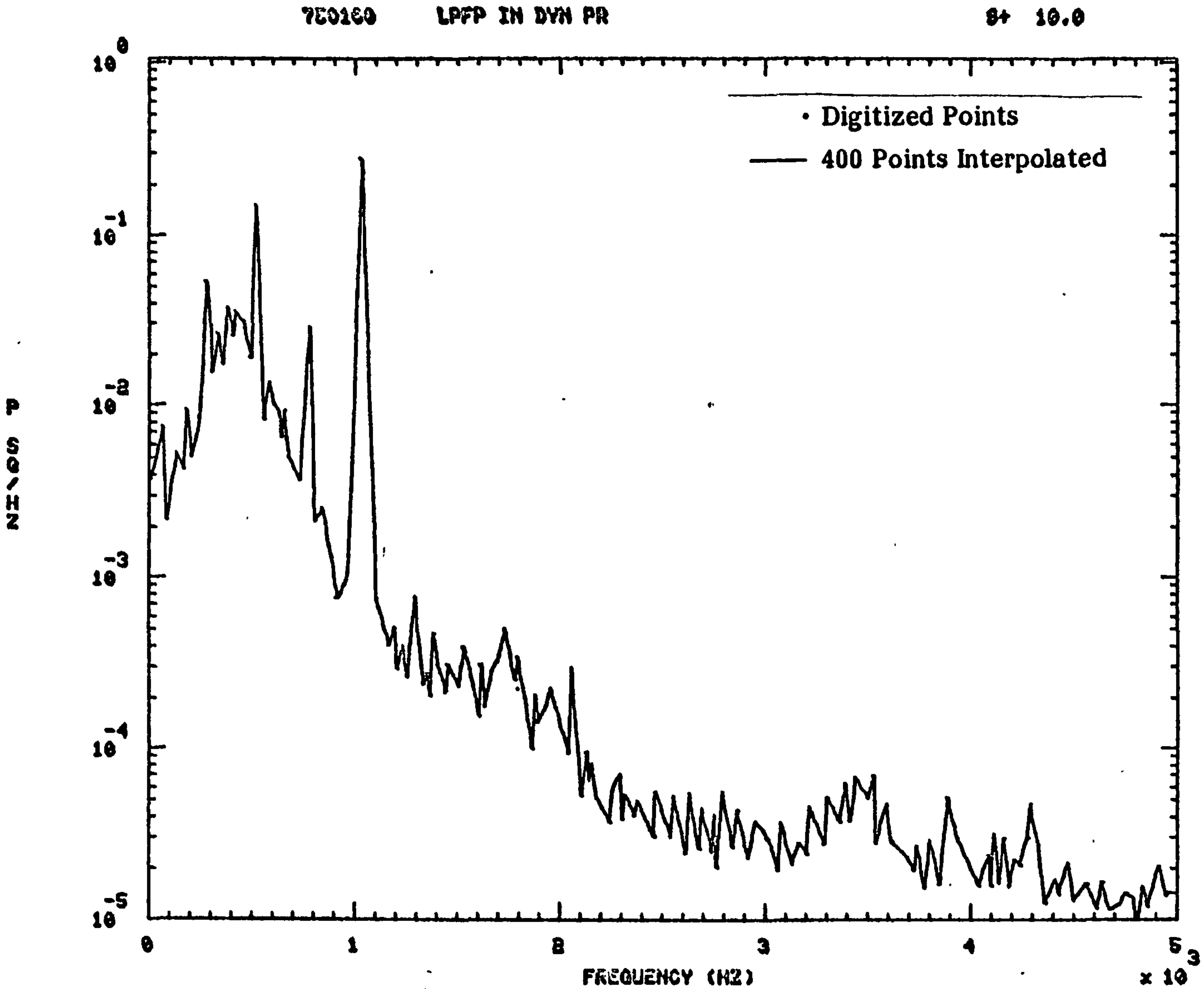


FIGURE 3-5. COMPARISON OF DIGITIZED POINTS TO 400-LINE SPECTRA OBTAINED USING THE CV400 SUBROUTINE

3.2 Data Filing System

The data base management system is designed to accommodate two types of data: time histories and PSD or other frequency domain data. Each time history file consists of two real numbers in each record. The first value in each record is a time and the second value is the magnitude of the quantity at that time. The data points are in chronological order. The file may consist of any number of data points (records), but most of the analysis program presently has an upper limit of 500 points.

Rapid access to time history data is gained through a file-naming structure that is programmed into the data base management system. Users of the data base management system and its associated subroutines need not understand the file naming structure; however, to illustrate the flexibility of the filing system and to assist those attempting to expand or modify the system, the naming structure is outlined here:

File Name: TXYYYYYY.ZZZ

where

T	- All time history files start with the letter T. PSD files never start with the letter T.
X	- May be any letter or number and designates the type of filter used in reducing the time history data. A description of the filter corresponding to each letter/number is contained in a directory file named DFILT.
YYYYYY	- The 6-digit test number.
ZZZ	- A 3-digit channel number. Channel descriptors corresponding to each channel number is contained in a directory file named DCHN.

Each PSD file consists of a single real number in each record. The first N records in the file contain the magnitude at N evenly spaced frequencies. The last 10 records contain information as follows:

Record	Contents
N+1	N - the number of magnitude points.
N+2	Composite on plots.*
N+3	Composite computed from digitized data.
N+4	Composite computed from data after being converted to N discrete points.*
N+5	Number of digitized points that could not be used in the conversion to N discrete points.*
N+6	Time slice.
N+7	Maximum frequency.
N+8	Spare.
N+9	Spare.
N+10	Spare.

The number of discrete points, N, may be any value, but most of the analysis programs currently have an upper limit of 512.

Access to PSD data files is gained through a file-naming structure similar to that used by the time histories. PSD files are named as follows:

File name: XXYYYYYY.ZZZ

where

XX - is a single letter followed by a single number. The letters A-S and U-Z and numbers 0-9 are used. The letter T is reserved for time histories. This letter combination designates the maximum frequency and time slice of the PSD contained in that file. The time slice and maximum frequency corresponding to each XX is found in a directory file named DPSD.

YYYYYY - The 6-digit test number.

ZZZ - A 3-digit channel number. Channel descriptors corresponding to each channel number is contained in a directory file named DCHN.

*Applies only to data obtained by digitizing plots.

3.3 Directory Files

A total of four directory files are used by the data base management system. Three of these files--DFILT, DCHN, and DPSD--have already been mentioned in conjunction with the file-naming structure. A fourth file, named TDIR, contains the engine number, HPOP number, HPFP number, and the test date associated with each test number. Since the use of these directory files is programmed into the data base management system, the user need not be concerned with their use. To develop new programs, however, it may be necessary to know the format of the files. The format of each directory file is:

TDIR - The file TDIR contains one record for each test number. Each record has the following format:

Column	Contents
1-6	Test number
8-16	Test date (YR:MO:DY)
18-26	Engine number
28-36	HPFP number
38-46	HPOP number

DCHN - The file DCHN contains a record for each channel descriptor. The position of the channel descriptor (i.e., record number) is the channel numbers. NOTE: Record must not be deleted from or inserted into this file. Each record has the following format:

Column	Contents
1-40	Channel descriptor

DFILT - The file DFILT contains a record for each filter description. Each record has the following format:

Column	Contents
1	Unique letter or number for filter description
3-15	Filter description

DPSD - The file PSD contains one record for each PSD in the data base. The format of each record is as follows:

Column	Contents
1-6	Test number
8-10	Slice code (letter number combination used in naming PSD files)
12-15	Channel number (must include high order zeros)
16-29	Slice time
31-44	Maximum frequency

3.4 Subroutines

Subroutines that have been developed for use with the SSME data base are stored in FORTRAN code in a file named MSFCSUBS.FTN. A file named MSFCSUBS.OBS contains the assembly code for these subroutines, which are ready for use when establishing a program.

To use these subroutines, simply call them in the program as required. The subroutine source code need not be included in the program. The following is a description of the subroutines included in MSFCSUBS.FTN.

Subroutine SCHN

Purpose: Search the channel descriptor file DCHN for a given channel descriptor and return with the corresponding channel number.

Subroutines and functions used: None.

Logical units used: DCHN is opened to LU 14 in SRO mode and closed before return. Error messages are returned as LU 11, which must be assigned before the call.

Usage: Call SCHN (ICH, CH, ISTAT)

ICH - ASCII representation of channel descriptor in 10A4 format. ICH must be dimensioned in the main program to be 10 words long.

CH - ASCII representation of the channel number in A3 format.

ISTAT - Status on return: 0 = normal return; -1 = unable to find given channel descriptor--the message "CHANNEL DESCRIPTOR NOT FOUND" outputted to LU 11.

Errors:

Open Error - An error in opening DCHN results in the message "OPEN ERROR XXX ON DCHN" being outputted to LU 11 where XXX is the open error code.

Code	Meaning
0	No error; requested function complete.
1	Illegal function.
2	LU error; illegal LU.
3	Volume error; no such volume.
4	Name error; no such name on given volume.
5	Size error; no room on disc for data or index block.
6	Protect error; mismatch on protection keys.
7	Privilege error; requested privilege may not be granted.
8	Buffer error; no room for FCB or buffer.
9	Assignment error; LU already assigned or off-line.
10	File descriptor syntax error.
12	Attempt to assign a trap generating device.
128-255	I/O error.

An open error results in the program being stopped.

Close Error: An error in closing DCHN results in the message "CLOSE ERROR XXX ON DCHN" being outputted to LU 11 where XXX is the close error code.

Code	Meaning
0	No error; requested function complete.
2	LU error.
9	Assignment error; LU not assigned.
128-255	I/O error.

A close error results in the program being stopped.

Read Error: A read error results in the message "READ ERROR ON SCHN" being outputted to LU 11 and the program is stopped.

Subroutine SPSD

Purpose: Search the PSD directory file DPSD of a given test number, channel number, slice time, and maximum frequency, and return with the slice code.

Subroutines and functions used: None.

Logical units used: The file DPSD is opened to LU 13 in SRO mode and closed before return. Error messages as outputted to LU 11, which must be assigned before the call.

Usage: Call SPSD (TEST, CH, SLICE, FREQM, ISLICE, ISTAT).

TEST	-	ASCII representation of the 6-digit test number in A4.A2 format. TEST must be dimensioned at least two words long.
CH	-	ASCII representation of the channel number in A3 format.
SLICE	-	Slice time (real variable).
FREQM	-	Maximum frequency (real variable).
ISLICE	-	ASCII representation of the slice code on return in A2 format.
ISTAT	-	Status on return: 0 = normal return; -1 = entry in DPSD matching request was not found.

Errors:

Open Error - An error in opening DPSD results in the message "OPEN ERROR XXX ON DPSD" being outputted to LU 11 where XXX is the open error code.

Code	Meaning
0	No error; requested function complete.
1	Illegal function.
2	LU error; illegal LU.
3	Volume error; no such volume.
4	Name error; no such name on given volume.
5	Size error; no room on disc for data or index block.
6	Protect error; mismatch on protection keys.
7	Privilege error; requested privilege may not be granted.
8	Buffer error; no room for FCB or buffer.
9	Assignment error; LU already assigned or off-line.
10	File descriptor syntax error.
12	Attempt to assign a trap generating device.
128-255	I/O error.

An open error results in the program being stopped.

Close Error: An error in closing DPSD results in the message "CLOSE ERROR XXX ON DPSD" being outputted to LU 11 where XXX is the close error code.

Code	Meaning
0	No error; requested function complete.
2	LU error.
9	*Assignment error; LU not assigned.*
128-255	I/O error.

A close error results in the program being stopped.

Read Error: A read error results in the message "READ ERROR ON SPSD" being outputted to LU 11 and the program is stopped.

Subroutine DSORT

Purpose: Sort vectors X and Y in descending order in X.

Subroutines and Function Used: None.

Logical Units Used: None.

Usage: CALL DSORT (X, Y, LEN)

X - REAL*4 vector.

Y - REAL*4 vector.

LEN - INT*4 length of vectors X and Y.

Errors: None.

Subroutine PEAK

Purpose: Find the peaks in a set of data and return with the X and Y values of each peak.

Subroutines Used: None.

Logical Units Used: None.

Usage: Call PEAK (X, Y, LEN, XP, YP, LENP).

X - REAL*4 X values in descending order.

Y - REAL*4 Y values.

LEN - INT*4 length of X and Y.

XP - REAL*4 X values of the peaks on return. XP must be dimensioned in calling program at least LEN/2 words long.

YP - REAL*4 Y values of the peaks on return. YP must be dimensioned in the calling program at least LEN/2 words long.

LENP - INT*4 number of peaks found.

Errors: None.

Subroutine STDIR

Purpose: Search the test number directory TDIR for a given combination of test number, test date, engine number, HPFP numbr, and HPOP number. Any parameter(s) may be a "DON'T CARE" by calling the subroutine with blanks in that field. On return, all fields will be filled if search is successful. The subroutine may be called repeatedly to find all entries in TDIR that satisfy the request.

Subroutines and Functions Called: None.

Logical Units Used: The file TDIR is opened to LU 13 the first time the subroutine is called and is closed when a call results in an unsuccessful search. Error messages are outputted on LU 11, which must be assigned before the call.

Usage: Call STDIR (TEST, DATE, ENG, HPFP, HPOP, ISTAT).

TEST - ASCII representation of the 6-digit test number in A4.A2 format.

DATE - ASCII representation of the test date (MO:DY:YR).

ENG - ASCII representation of the engine number in 2A4.A2 format.

HPFP - ASCII representation of the HPFP number in 2A4.A2 format.

HPOP - ASCII representation of the HPOP number in 2A4.A2 format.

ISTAT - Call status:

On first call - ISTAT not equal 0.
Call to continue search - ISTAT equal 0.
On successful return - ISTAT equal 1.
On unsuccessful return - ISTAT equal 0.
On file read error - ISTAT equal 1.

Page Intentionally Left Blank

Errors:

Open Error - An error in opening TDIR results in the message "OPEN ERROR NO XXX" being outputted to LU 11 where XXX is the open error code.

Code	Meaning
0	No error; requested function complete.
1	Illegal function.
2	LU error; illegal LU.
3	Volume error; no such volume.
4	Name error; no such name on given volume.
5	Size error; no room on disc for data or index block.
6	Protect error; mismatch on protection keys.
7	Privilege error; requested privilege may not be granted.
8	Buffer error; no room for FCB or buffer.
9	Assignment error; LU already assigned or off-line.
10	File descriptor syntax error.
12	Attempt to assign a trap generating device.
128-255	I/O error.

An open error results in the program being stopped.

Subroutine RDTH

Purpose: Read the time history file specified by test number, filter code, and channel number, and return with the contents.

Subroutines and Functions Used: None.

Logical Units Used: The specified time history file is opened to LU 12 and closed before return error messages are outputted to LU 11, which must be assigned before the call.

Usage: Call RDTH (TEST, CH, IFLT, TIME, MAG, LEN, ISTAT).

TEST - ASCII representation of the test number in A4.A2 format.

CH - ASCII representation of the channel number in A3 format.

IFLT - ASCII representation of the filter code in A1 format.

TIME - REAL*4 time values on return.

MAG - REAL*4 magnitude values on return

LEN - Dimension of vectors TIME and MAG on CALL number of data points read on return.

ISTAT - 0 = Normal return

-1 = Number of data points greater the LEN at CALL

1 = Illegal function

2 = LU error; illegal LU

3 = Volume error; no such volume

4 = Name error; no such mane on given volume

5 = Size error; no room on disc for data or index block

6 = Protect error; mismatch on protection keys

7 = Privilege error; requested privilege may not be granted

8 = Buffer error; no room for FCB or buffer

9 = Assignment error; LU already assigned or off-line

11 = File descriptor syntax error

12 = Attempt to assign a trap generating device

128-255 = I/O error

Errors:

Read Error - A read error results in the message "Read Error on File XXX" being outputted to LU 11 where XXX is the file name being read.

Subroutine ERROR

Purpose: Output an error message on LV11 in the form:

Error message ISTAT = 10

Subroutines and Functions Used: None.

Logical Units Used: The message is outputted to LU 11, which must be assigned before the call.

Usage: Call ERROR (MESG, ISTAT).

MESG - May be a 1X5 array with ASCII representation of the message or the message may be placed in the call CALL ERROR ("MESSAGE GOES HERE: ISTAT) the message must be 20 characters long including blanks ISTAT - INT*4 variable or constant to be outputted. Example: ISTAT = 10; IF (ISTAT.NE.0) CALL ERROR ("THIS IS A SAMPLE ' ISTAT) will result in the following message being outputted to LU 11: THIS IS A SAMPLE ISTAT = 10.

Errors: None.

Subroutine PROMPT

Purpose: Output a message to LU11.

Usage: Call PROMPT(TEXT)

TEST - Message is in the form "EXAMPLE MESSAGE!" the message may be any length but must end with a !. Example: Call PROMPT ("EXAMPLE MESSAGE!") result in the following being outputted to LU 11: EXAMPLE MESSAGE!.

Errors: None.

Subroutine INTERP

Purpose: Interpolate between data points.

Usage CALL INTERP (X, Y, XX YY, SLOPE, LEN).

X - REAL*4 array containing x data points.

Y - REAL*4 array containing y data points.

XX - REAL*4 scalar the x value for which a y value is to be computed.

YY - REAL*4 scalar y value found to correspond to given x value.

SLOPE - REAL*4 scalar slope at XX.

LEN - The length of the vectors X and y.

Errors: None.

Subroutine RDPSD

Purpose: Read the contents of a PSD data file and put frequency values in X and magnitudes in Y.

Subroutines and Function Called: None.

Logical Units Used: The PSD data file is opened to LU 12 and closed before the return usage: Call RDPSD (TEST, CH, ISLICE, X, Y, COMP, LEN, ISTAT)

TEST	-	Test number in A4-A2 format.
CH	-	Channel number in A3 format.
ISLICE	-	Slice code in A2 format.
X	-	REAL*4 any containing frequency value on return must be dimensioned at least LEN words long in calling program.
Y	-	REAL*4 array containing magnitude values on return must be dimensioned at least LEN+10 words long in calling program. On return: Y(LEN+1) = LEN Y(LEN+2) = composite on plots Y(LEN+3) = composite computed from digitized data Y(LEN+4) = composite computed from discrete spectra data Y(LEN+5) = number digitized points not assigned during conversion to discrete spectra Y(LEN+6) = time slice Y(LEN+8)-Y(LEN+10) = spare
LEN	-	Dimension of X at call and number of data points read on return.
ISTAT	-	Call status on return

-625 to -755	I/O error at close
-509	Assignment error; LU not assigned at close
-502	LU error at close
-3	LEN check failed
-2	Data exceed dimension of X
-1	Read error
0	Normal return
1	Illegal function
2	LU error; illegal LU
3	Volume error; no such volume

4	Name error; no such name on given volume
5	Size error; no room on disc for data or index block
6	Protect error; mismatch on protection keys
7	Privilege error; requested privilege may not be granted
8	Buffer error; no room for FCB or buffer
9	Assignment error; LU already assigned or off-line
11	File descriptor syntax error
12	Attempt to assign a trap generating device
128-255	I/O error

Errors: None.

Subroutine SPSD1

Purpose: Search the PSD directory file for a given test number and channel number, and return with the time slice maximum frequency and slice code. The subroutine can be called repeatedly to find all entries that satisfy the request.

Subroutines and Function Called: None.

Logical Unit Used: The DPSD file is opened to LU 13 and remains open until a search is unsuccessful. Error messages are outputted on LU 11, which must be assigned before the call.

Subroutine CV400

Purpose: Converts digitized PSD data taken at nonuniform frequency intervals to 400 data points at constant frequency intervals.

3.5 Programs

Both data base management and data analysis programs have been written and used with the SSME data. The data base management program includes programs for entering new data, for summarizing the contents of the data base, and for removing erroneous or incorrectly identified data. Data analysis programs include programs for plotting data and computing statistcal data summaries. The following is a brief summary of the programs that have been completed to date.

Program name: DIGTIME

Purpose: Used in digitizing time history plots.

Method: The program asks the operator to enter the filter description, test number, and channel descriptor for the plot to be digitized. It then checks the TDIR directory file to see if the test number has already been entered. If the entry is found, the test date, engine number, HPOP number, and HPFP number is displayed for verification. If no entry is found, the operator is asked to enter the test date, engine number, HPOP number, and HPFP number, and this information is entered into TDIR.

The DCHN directory file is then searched for the channel descriptor and to obtain the corresponding channel number. If the channel descriptor is not found, then the program asks if it is a new channel descriptor. If the operator responds with "yes", then an entry is made in DCHN assigning to that channel descriptor a channel number. If the response is "no" then the operator is given the opportunity to reenter the channel descriptor.

The program then asks the operator to enter the X and Y values at the origin of the plot to be digitized and the values of one other point on the x-axis and one on the y-axis. It then instructs the operator to digitize these points using the digitizer pad. From this information the program determines the orientation of the plot on the digitizer pad

and the scaling parameter needed to convert the output of the digitizer pad into engineering units. The operator digitizes the plot in as many straight-line segments as is necessary to produce an accurate digital image. When the digitizing is complete, the program creates a file, using the naming structure described previously, and enters the data in that file, completing the process of entering a time history into the data base.

Program name: DIGPSD

Purpose: Used in digitizing PSD plots.

Method: The DIGPSD program starts by asking the operator for the test number and channel descriptor--much like DIGTIME--and, if necessary, makes the same search of entries in the TDIR and DCHN directory files. In addition, it asks for the time slice and maximum frequency of the PSD to be digitized. With this information, it searches the DPSD directory file to find a unique slice code for the plot to be digitized. The slice code, test number, and channel number are used to name the data file to be created. The program asks the operator to digitize three points on the axes of the plot from which it determines the plot orientation and scaling factors. The program assumes that the magnitude axis is logorithm and the frequency axis is linear. After the plot is digitized, the program converts the data to 400 magnitude points at uniform frequency intervals. These points are stored in a data file named using the file naming structure described previously. The program asks the operator to enter the composite value. It also computes the composite from the digitized data and the data after being converted to 400 points. All three composite values are entered into the data file, and the computed values are displayed so the operator can check the digitization. An entry is then made in the DPSD directory file, completing the data entry process.

Program name: REMTH

Purpose: Remove time history data from the data base.

Method: The program asks for the test number, filter description, and channel descriptor of the time history to be removed. It then reads the corresponding time history file and displays the minimum and maximum values of magnitude and time as well as the number of data points in the file. The operator is then asked, by the program, if this is the file to be removed. If the operator's response is yes, then the file is removed; otherwise it remains unchanged.

Program name: REMPSD

Purpose: Remove PSD data from the data base.

Method: The program asks the operator to enter the test number, channel descriptor, time slice, and maximum frequency of the PSD data to be removed. It then searches the DPSD directory file for the given combination of test number, channel descriptor, time slice, and maximum frequency to obtain the slice code. The data file is then read. The minimum and maximum magnitudes and composite value is displayed. The operator is then asked if this is the PSD data to be removed. A response of yes by the operator results in the data file being deleted and the corresponding entry in the DPSD directory file being removed. Any other response results in no action being taken.

Program name: LSTTH

Purpose: List the test number and channel descriptor of all time history data in the data base.

Method: The program uses a system command to obtain a list of all the time history file names from which the test numbers and channel numbers are obtained. The DCHN directory file is read to obtain the channel descriptor corresponding to each channel number. The test number and channel descriptor for each data file is then printed.

Program name: LSTPSD

Purpose: List the test number, time slice, maximum frequency, and channel descriptor of each PSD in the data base.

Method: The DPSD directory file is read to obtain the test number, time slice, maximum frequency, and channel number of each PSD in the data base. The DCHN directory file is used to find the channel descriptor that corresponds to each channel number. The test number, time slice, maximum frequency, and channel descriptor is then printed.

Program name: PLOTTH

Purpose: Plot time history data.

Method: The program asks for the test number filter description and channel descriptor. The channel number corresponding to the given channel descriptor is found by searching the DCHN directory file. The data file is read and the data is plotted on a Tektronix graphics terminal from which a hard copy may be produced. The plot includes scaled and labeled axes and a title. The title consists of the test number, channel descriptor, and filter description.

Program name: PLOTPSD

Purpose: Plot PSD data.

Method: The program asks the operator to enter the test number, channel descriptor, time slice, and maximum frequency of the PSD data to be plotted. The program searches the DCHN directory file to find the channel number and the DPSD directory file for the slice code. The corresponding data file is read and the data is plotted on a Tektronix graphics terminal. The axes are scaled and labeled, and a title consisting of the test number, channel descriptor, and slice time is included on the plot.

Program name: STATPSD

Purpose: For a set of PSDs specified by engine number, HPFP number, HPOP number (any or all may be "don't care"), maximum frequency, channel descriptor, and thrust range, calculates the following as a function of frequency.

- Mean

- Standard deviation
- rms
- Third central moment
- Fourth central moment
- Minimum value
- Maximum value
- Range

Method: The program asks the operator to enter the engine number, HPOP number, and HPFP number of the data to be considered. Entries that are left blank are assumed to be "don't care." The program searches the TDIR directory file to find the test numbers that correspond to the given combination of engine number, HPOP number, and HPFP number. The DCHN directory file is searched to find the channel number corresponding to the channel descriptor. The DPSD file is searched to find slice codes and time slices that correspond to the given channel number, maximum frequency, and each test number. For each entry found, the corresponding thrust time history is read to find the thrust at that time slice. If the thrust is within the specified range, then the PSD data file is read and included in the statistical calculations. Once all the PSDs that correspond to the given set of requirements have been found, the data is outputted to a file in a format to be used by other programs. This output file also includes information needed to title and label the axis of plots.

Program name: EXPSD

Purpose: For each PSD as specified by engine number, HPFP number, HPOP number (any or all may be "don't care"), maximum frequency, and channel descriptor, finds the magnitude and frequency of the N largest peaks, where N is specified by the operator. Also, finds the thrust that corresponds with each PSD considered. The output of this program can be used to plot the magnitude or frequency of a given peak as a function of thrust.

Method: PSD files that satisfy the specification are found by using the method described for STATPSD. The peaks in each PSD are found by examining the slope of the PSD as a function of frequency. When the slope changes from positive to negative, that point is stored as a peak. When all the peaks in a PSD are found, they are sorted by magnitude, and the N largest peaks are reported to an output file together with the thrust corresponding to that PSD. This process is repeated for each PSD that satifies the given specification.

Program name: PLOT

Purpose: Plot the contents of a file generated by analysis programs such as STATPSD and EXPSD.

Method: The data file to be plotted is specified by name when the program is started. From this file, the program reads a description of each channel of data that the file contains. These descriptions are displayed and the operator is asked to enter a number corresponding to the channel to be plotted on the x-axis and for the channel to be plotted on the y-axis. The operator is asked to specify linear or logarithmic scaling for each axis. The choice of line plot or dots is given. If line plot is specified, the option of sorting the data by x-values is given. The plots are titled and axes are labeled using the information provided by the data file.

Program name: LOADTHST

Purpose: Enter thrust profile or rms time history data into the data base via 9-track magnetic tape.

Method: The program reads the test number, engine number, HPOP number, HPFP number, and test data from the tape. The TDIR file is checked for the test number. If the test number is not found, the test number, HPOP number and HPFP number are added to the file. The program then reads the time history from the tape and looks to see whether the information is already in the data base. If it is not in the data base, then the history read from the tape is added to the data base.

Program name: LOADPSD

Purpose: Enter PSD data into the data base.

Method: The program reads the test number channel descriptor and time slice from the tape. It then checks the TDIR file to see if that test number is present. If not found, the program prints an error message and proceeds to the next data set on the tape. The program then checks the DPSD file for a data file with the same test number, channel descriptor, and time slice. If one is found, the program proceeds to the next data set on the tape; if no match is found, a slice code is obtained and the PSD data is transferred from tape to the data base management system.

A flow diagram of the data base management and analysis software is illustrated in Figure 3-6.

A computer listing of the programs and subroutines discussed in this section is provided on the following pages. This code is operational on PC/compatible systems. The routines are also available on disc for efficient transfer to MSFC computers.

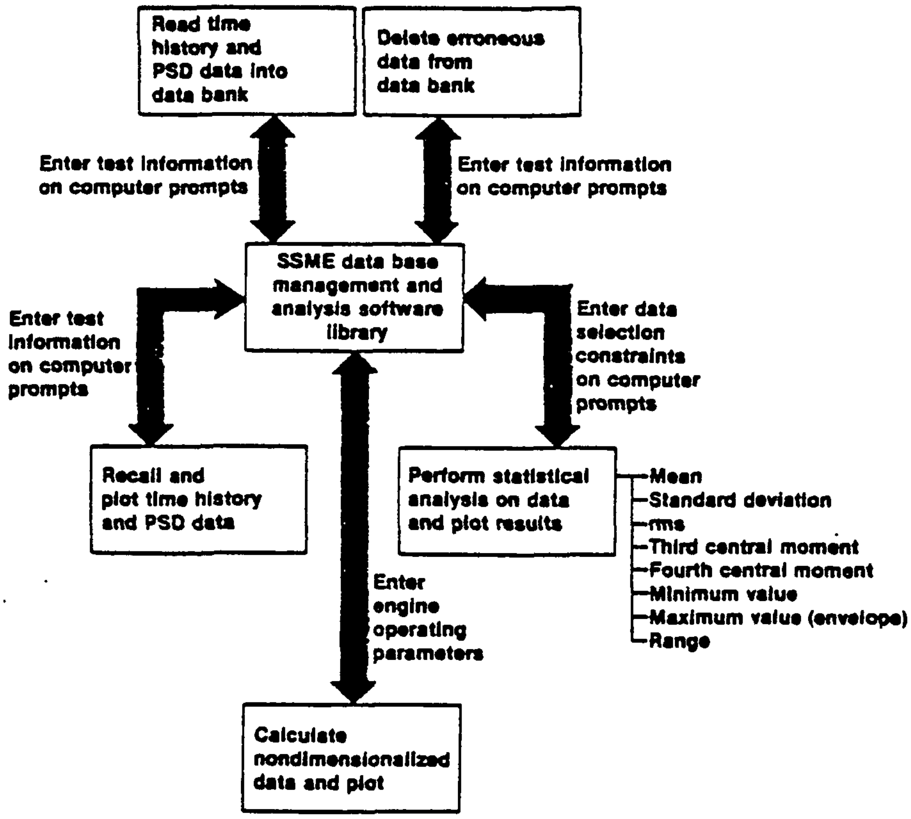


FIGURE 3-6. SSME DATA BASE MANAGEMENT AND ANALYSIS SOFTWARE OPERATIONAL SEQUENCE AND CAPABILITIES

TABLE 3-1

DATA BASE ANALYSIS PROGRAM LISTING


```
D Line# 1      7                                   Microsoft FORTRAN77 V3.20 02/84
      $DEBUG
      C       MAIN PROGRAM TO CALCULATE STATISTICS ON PSD'S
      C       THE PROGRAM SEARCHES THE DATA BASE FOR A GIVEN COMBINATION
      C       OF ENGINE NO., HPFP NO., HPOP NO. (ANY OR ALL MAY BE "DON'T CARE")
      C       CHANNEL DESCRIPTOR AND THRUST RANGE.
      C
      C       THE MEAN,STANDARD DEVIATION,THIRD CENTRAL MOMENT,FOURTH CENTRAL
      C       MOMENT,RMS,MINIMUM VALUE,MAXIMUN VALUE AND RANGE ARE CALCULATED
      C
      C       OUTPUT DATA IS WRITTEN ON LU 7 (120 CHARACTORS)
      C       THE GIVEN ENGINE NUMBER, HPFP NUMBER AND HPOP NUMBER, TEST NUMBERS
      C       PSD'S AND QUANTITY OF PSD FROM EACH TEST USED ARE OUT PUT ON LU 5
      C
      C
      C
              CHARACTER TEST*6, DATE*8, ENG*10, HPFP*10, HPOP*10
              CHARACTER TESTX*6, DATEX*8, ENGX*10, HPFPX*10,HPOPX*10
              CHARACTER ICH*20,CH*3
              REAL TIME(500),MAG(500),X(500),Y(500)
              REAL XXFOUR(200)
              REAL XXTOT(200),XXMIN(200),XXMAX(200),XXSQR(200),XXCUB(200)
              OPEN (7,FILE='DATA.PRT')
              OPEN (5,FILE='STATS.DAT')
      C
      C       INITIALIZE VARIABLES
      C
              DO 5 J=1,200
1             XXTOT(J)=0.0
1             XXMIN(J)=1.0E32
1             XXMAX(J)=-1.0E32
1             XXCUB(J)=0.0
1             XXFOUR(J)=0.0
1        5    XXSQR(J)=0.0
              ISTAT=0
              ISTATT=0
         20   FORMAT (A10)
         22   FORMAT (' 'A10)
         30   FORMAT (A20)
              TEST='      '
         10   DATE='        '
      C
      C       ASK FOR ENGINE NO., HPFP NO., HPOP NO.
      C
         1    WRITE (*,40)
         40   FORMAT (' ENTER ENGINE NO.')
              READ (*,20)ENG
              IF(ENG.EQ.'STOP')STOP
              WRITE (*,50)
         50   FORMAT (' ENTER HPFP NO. ')
              READ (*,20)HPFP
              WRITE (*,60)
         60   FORMAT (' ENTER HPOP NO. ')
              READ (*,20)HPOP
      C
      C       ASK FOR CHANNEL DESCRIPTOR AND FIND CHANNEL NO.
      C
```

```
      WRITE(*,*)'ENTER CHANNEL DESCRIPTER!'
      READ (*,30)ICH
      CALL SCHN(ICH,CH,ISTAT)
```



```
      IF(ISTAT.EQ.-1) GO TO 1
      IF(ISTAT.NE.0)CALL ERROR(' IN ICHN                ',ISTAT)
      CALL DPSDTEMP(TESTX,CH,ISTAT)
C
C     ASK FOR MAXIMUM FREQUENCY
C
      WRITE (*,*)'ENTER MAX. FREQUENCY!'
      READ (*,21)FREQM
      AFREQS=FREQM/400.
21    FORMAT(G13.6)
      FREQL=FREQM
C
C     ASK FOR THRUST RANGE
C
      WRITE(*,*)'ENTER LOWER THRUST LIMIT!'
      READ (*,21)THRL
      WRITE(*,*)'ENTER UPPER THRUST LIMIT!'
      READ (*,21)THRH
C
C     WRITE NUMBER OF OUT PUT COLUMNS TO LU 7 (THIS CASE 9)
C
      WRITE(7,64)
64    FORMAT(' 0009')
C
C     WRITE HEADER
C
      WRITE(7,65)ICH,THRL,THRH
65    FORMAT(' 'A10,5X,'THRUST RANGE = 'F5.1,' - 'F5.1)
C
C     WRITE OUTPUT COLUMN ID'S
C
      WRITE(7,66)
66    FORMAT(' FREQUENCY  HZ'/' MEAN  P**2/HZ'
     1/' STANDARD DIVIATION  P**2/HZ'/' THIRD CENTRAL MOMENT P**6/HZ'
     4/' FOURTH CENTRAL MOMENT  P**8/HZ'
     2/' R M S   P**2/HZ'/' MINIMUM VALUE P**2/HZ'
     3/' MAXIMUM VALUE  P**2/HZ'/' RANGE  P**2/HZ')
C
C     WRITE ENGINE NO., HPFP NO., HPOP NO. COMBO TO LU 5
C
      WRITE(5,67)ENG
      WRITE(5,68)HPFP
      WRITE(5,69)HPOP
67    FORMAT(' ENGINE NO. = 'A10)
68    FORMAT('    HPFP NO. = 'A10)
69    FORMAT('    HPOP NO. = 'A10//' TEST NO.  QTY.')
      JK=0
C
C     SEARCH FOR TEST NO. FOR GIVEN COMBO
C
70    CONTINUE
      TESTX=TEST
71    DATEX=DATE
      ENGX=ENG
      HPOPX=HPOP
72    HPFPX=HPFP
```

```
      CALL STDIR (TESTX, DATEX, ENGX, APCPX, APOPX, ISTATT)
      IF (ISTATT.LT.0) GO TO 990
      IF (ISTATT.EQ.0) GO TO 300
```



```
        NCOUT=0
        LENT=100
  C
  C     READ THRUST DATA
  C
  C      CALL RDTH(TESTX,'001 ','1    ',TIME,MAG,LENT,ISTAT)
  C      IF(ISTAT.EQ.4)GO TO 1100
  C      IF(ISTAT.NE.0)CALL ERROR('IN RDTH      ',ISTAT)
  C
        ISTAT1=1
  C
  C     SEARCH FOR PSD DATA
  C
  100   CALL SPSD1(TESTX,CH,SLICE,FREQ,ISLICE,ISTAT1,THRUST)
  C
  C     IF FILE FOUND GO TO 110
  C
        IF(ISTAT1.NE.-1)GO TO 110
  C
  C     IF NO MORE FOUND WRITE TEST NO. AND QUANTITY OF PSD'S FOUND
  C     FOR THAT TEST NO...   GET NEW TEST NO.
  C
        IF(NCOUT.EQ.0)GO TO 70
        WRITE(*,105)TESTX,NCOUT
        WRITE(5,105)TESTX,NCOUT
  105   FORMAT('  'A6,2X,I3)
        GO TO 70
  C
  110   CONTINUE
  C
  C     IF OUT SIDE THRUST RANGE GET NEW PSD
  C
        IF((THRUST.LT.THRL).OR.(THRUST.GT.THRH))GO TO 100
  C
  C     IF PSD DOES NOT COVER FREQUENCY RANGE GET NEW PSD
  C
        IF(FREQ.NE.FREQL)GO TO 100
        NCOUT=NCOUT+1
        LEN=500
  C
  C     READ PSD DATA FILE
  C
        CALL RDPSD(TESTX,CH,ISLICE,X,Y,COM,LEN,ISTAT)
        IF(ISTAT.NE.0)CALL ERROR('IN RDPSD     ',ISTAT)
        JK=JK+1
        DO 200 J=1,400
1       FREQZ=(J-1)*AFREQS
1 C
1 C     ACCUMULATE DATA FOR 400 FREQUENCIES BETWEEN 0 AND MAX FREQUENCY
1 C
1       XXTOT(J)=XXTOT(J)+Y(J)
1       XXMIN(J)=AMIN1(XXMIN(J),Y(J))
1       XXMAX(J)=AMAX1(XXMAX(J),Y(J))
1       XXSQR(J)=XXSQR(J)+Y(J)**2
1       XXFOUR(J)=XXFOUR(J)+Y(J)**4
1       XXCUB(J)=XXCUB(J)+Y(J)**3
  200   CONTINUE
```

```
1   200    CONTINUE
           GO TO 100
    300    CONTINUE
```



```
           IF(JK.GT.0)GO TO 301
           WRITE(*,*)'NONE FOUND!'
           STOP
      C
      C    CALCULATE AND OUTPUT STATISTICS
      C
    301    DO 400 J=1,400
1          FREQS=(J-1)*AFREQS
1          XMEAN=XXTOT(J)/JK
1          XM2=XXSQR(J)/JK
1          VAR=XM2-XMEAN**2
1          STD=SQRT(ABS(VAR))
1          RMS=SQRT(XM2)
1          RANGE=XXMAX(J)-XXMIN(J)
1          XM3=XXCUB(J)/JK
1          XM4=XXFOUR(J)/JK
1          XU3=XM3-3.*XMEAN*XM2-2.*XMEAN**3
1          XU4=XM4-4.*XM3*XMEAN+6.*XM2*XMEAN**2-3.*XMEAN**4
1   400    WRITE(7,310)FREQS,XMEAN,VAR,STD,XU3,RMS,XXMIN(J),XXMAX(J)
1         1,RANGE
    310    FORMAT(' '9G13.6)
           WRITE(5,320)JK
    320    FORMAT(' TOTAL      'I3)
           STOP
    990    WRITE (*,991)
    991    FORMAT (' FILE READ ERROR')
           STOP
    1100   WRITE(*,1110)TESTX
    1110   FORMAT(' THRUST DATA FOR TEST NO. 'A4,A2
          1' NOT IN DATA BASE')
           GO TO 70
           END
```

Name	Type	Offset	P	Class
ABS				INTRINSIC
AFREQS	REAL	13000		
AMAX1				INTRINSIC
AMIN1				INTRINSIC
CH	CHAR*3	12986		
COM	REAL	13640		
DATE	CHAR*8	12859		
DATEX	CHAR*8	13556		
ENG	CHAR*10	12891		
ENGX	CHAR*10	13564		
FREQ	REAL	13610		
FREQL	REAL	13012		
FREQM	REAL	12996		
FREQS	REAL	13648		
FREQZ	REAL	13644		
HPFP	CHAR*10	12924		
HPFPX	CHAR*10	13584		
HPOP	CHAR*10	12956		
HPOPX	CHAR*10	13574		
ICH	CHAR*20	12966		
ISLICE	INTEGER*4	13614		

```
ISTAT   INTEGER*4      12820
ISTAT1  INTEGER*4      13602
ISTATT  INTEGER*4      12824
```



```
J       INTEGER*4      12816
JK      INTEGER*4      13552
LEN     INTEGER*4      13636
LENT    INTEGER*4      13598
MAG     REAL           10816
NCOUT   INTEGER*4      13594
RANGE   REAL           13672
RMS     REAL           13668
SLICE   REAL           13606
SQRT                            INTRINSIC
STD     REAL           13664
TEST    CHAR*6         12853
TESTX   CHAR*6         12989
THRH    REAL           13020
THRL    REAL           13016
THRUST  REAL           13618
TIME    REAL              16
VAR     REAL           13660
X       REAL            2016
XM2     REAL           13656
XM3     REAL           13676
XM4     REAL           13680
XMEAN   REAL           13652
XU3     REAL           13684
XU4     REAL           13688
XXCUB   REAL           10016
XXFOUR  REAL            6016
XXMAX   REAL            9216
XXMIN   REAL            7616
XXSQR   REAL            8416
XXTOT   REAL            6816
Y       REAL            4016
```

```
C
C$PROG SCHN
C       PURPOSE:  SEARCH THE DESCRIPTOR FILE FOR A GIVEN DESCRIPTOR
C                 AND RETURN THE CHANNEL NUMBER
C
        SUBROUTINE SCHN(ICH,CH,ISTAT)
C
C       ICH    - DISCRIPTOR (20 CHARACTERS)
C       CH     - CHANNEL NO.  (RETURNED)
C       ISTAT  - RETURN STATUS
C                0-NORMAL
C               -1-DISCRIPTOR NOT FOUND
        CHARACTER ICHY*4
        CHARACTER FD14*4,CH*3,ICH*20,ICHX*20,DMY*1
        DATA FD14/'DCHN'/
        OPEN(14,FILE=FD14,IOSTAT=ISTAT)
        IF(ISTAT.NE.0)GO TO 2040
        I=0
  3000  I=I+1
        READ(14,703,END=3500,ERR=1000)ICHX
  703   FORMAT(A20)
        IF(ICHX.NE.ICH) GO TO 3000
```

```
 3025 CLOSE(14,IOSTAT=IERR)
      IF(IERR.NE.0)GO TO 2060
      I=I+1000
```



```
      WRITE(ICHY,3030)I
 3030 FORMAT(I4)
      GO TO 3600
 3500 WRITE(*,3501)
 3501 FORMAT(' CHANNEL DESCRIPTOR NOT FOUND')
      ISTAT=-1
      CLOSE(14,IOSTAT=IERR)
      IF(IERR.NE.0)GO TO 2060
      RETURN
 3600 READ(ICHY,701)DMY,CH
 701  FORMAT(A1,A3)
      WRITE(*,701)DMY,CH
      RETURN
 1000 WRITE(*,1110)
 1110 FORMAT(' READ ERROR ON SCHN')
      STOP
 2040 WRITE(*,2050)ISTAT
 2050 FORMAT(' OPEN ERROR 'I3'ON DCHN')
      STOP
 2060 WRITE(*,2070)IERR
 2070 FORMAT(' CLOSE ERROR 'I3' ON DCHN')
      STOP
      END
```

Name	Type	Offset	P	Class
CH	CHAR*3	4	*	
DMY	CHAR*1	13904		
FD14	CHAR*4	13824		
I	INTEGER*4	13828		
ICH	CHAR*20	0	*	
ICHX	CHAR*20	13832		
ICHY	CHAR*4	13862		
IERR	INTEGER*4	13858		
ISTAT	INTEGER*4	8	*	

```
C$PROG SPSD
C     PURPOSE:  SEARCH THE PSD DIRECTORY FOR A GIVEN TEST NO.
C               CHANNEL NO.,TIME SLICE,AND MAX FREQ.- -- RETURN
C               WITH THE SLICE CODE
C NOT VERIFIED ON IBM 12/85
C
      SUBROUTINE SPSD(TEST,CH,SLICE,FREQM,ISLICE,ISTAT)
      CHARACTER TEST*6,TEST1*6
      CHARACTER FD13*4
      DATA FD13 /'DPSD'/
      OPEN(13,FILE=FD13,IOSTAT=ISTAT)
      IF(ISTAT.NE.0) GO TO 1180
 820  READ(13,830,END=860,ERR=1000)TEST1,ISLICE,CHX,SLICE1,
     1FREQ
 830  FORMAT(A6,1X,A2,1X,A3,2(1X,G13.6))
      IF(TEST.NE.TEST1)GO TO 820
 840  CONTINUE
      IF(CHX.NE.CH)GO TO 820
      IF(SLICE.NE.SLICE1)GO TO 820
```

```
          IF(FREQ.NE.FREQM)GO TO 820
     850  CLOSE(13,IOSTAT=IERR)
          IF(IERR.NE.0)GO TO 2000
```



```
          RETURN
     860  ISTAT=-1
          GO TO 850
    1000  WRITE(*,1110)
    1110  FORMAT(' READ ERROR IN SPSD')
          STOP
    1180  WRITE(*,1190)ISTAT
    1190  FORMAT(' OPEN ERROR NO. 'I3,' ON FILE DSPD')
          STOP
    2000  WRITE(*,2010)IERR
    2010  FORMAT(' CLOSE ERROR NO. 'I3'ON FILE DPSD')
          STOP
          END
```

Name	Type	Offset	P	Class
CH	REAL	4	*	
CHX	REAL	14006		
FD13	CHAR*4	13996		
FREQ	REAL	14014		
FREQM	REAL	12	*	
IERR	INTEGER*4	14046		
ISLICE	INTEGER*4	16	*	
ISTAT	INTEGER*4	20	*	
SLICE	REAL	8	*	
SLICE1	REAL	14010		
TEST	CHAR*6	0	*	
TEST1	CHAR*6	14000		

```
      C$PROG DSORT
      C
      C       PURPOSE:  SORT X AND Y IN TO DECENDING ORDER IN X
      C   NOT VERIFIED ON IBM 12/85
      C
              SUBROUTINE DSORT(X,Y,LEN)
              DIMENSION X(1),Y(1)
              LM1=LEN-1
              DO 10 I=1,LM1
1             LEND=LEN-I
1             DO 10 J=1,LEND
2             JP1=J+1
2             IF(X(J).GT.X(JP1))GO TO 10
2             XTEMP=X(JP1)
2             YTEMP=Y(JP1)
2             X(JP1)=X(J)
2             Y(JP1)=Y(J)
2             X(J)=XTEMP
2             Y(J)=YTEMP
2       10    CONTINUE
              END
```

Name	Type	Offset	P	Class
I	INTEGER*4	14156		
J	INTEGER*4	14168		

```
JP1      INTEGER*4      14176
LEN      INTEGER*4          8 *
LEND     INTEGER*4      14164
```



```
LM1      INTEGER*4      14152
X        REAL               0 *
XTEMP    REAL           14180
Y        REAL               4 *
YTEMP    REAL           14184
```

```
      C$PROG PEAK
      C      PURPOSE:  SEARCH PSD DATA TO FIND PEAKS
      C   NOT VERIFIED ON IBM 12/85
      C
             SUBROUTINE PEAK(X,Y,LEN,XP,YP,LENP)
      C
      C      X    - PSD FREQ POINTS
      C      Y    - PSD MAG.  POINTS
      C      LEN - NUMBER OF DATA POINTS
      C      XP    - FREQUENCY OF PEAKS
      C      YP   - MAGNITUDE OF PEAKS
      C      LENP - NUMBER OF PEAKS
      C
             DIMENSION X(1),Y(1),XP(1),YP(1)
             LENM1=LEN-1
             LENP=0
             DO 10 J=2,LENM1
1            JP1=J+1
1            JM1=J-1
1            IF(.NOT.(Y(J)-Y(JM1).GT.0.0).AND.(Y(J)-Y(JP1).LT.0.0))GO TO 10
1            LENP=LENP+1
1            XP(LENP)=X(J)
1            YP(LENP)=Y(J)
1       10   CONTINUE
             RETURN
             END
```

```
Name     Type          Offset P Class

J        INTEGER*4      14192
JM1      INTEGER*4      14204
JP1      INTEGER*4      14200
LEN      INTEGER*4          8 *
LENM1    INTEGER*4      14188
LENP     INTEGER*4         20 *
X        REAL               0 *
XP       REAL              12 *
Y        REAL               4 *
YP       REAL              16 *
```

```
      C
      C$PROG STDIR
      C      SUBROUTINE STDIR
      C
      C      PURPOSE:  SEARCH THE TEST NUMBER DIRECTORY FILE (TDIR) FOR A SPECI
      C      COMBINATION OF TEST NUMBER,ENGINE NUMBER,OXIDIZER PUMP NUMBER, FUE
      C      PUMP NUMBER AND TEST DATE
      C
```

```
C USAGE
C       CALL STDIR(TEST,DATE,ENG,HPFP,HPOP,ISTAT)
C
```



```
C       TEST - TEST NUMBER TO BE SEARCHED FOR (6* CHACRACTERS)
C       DATE - TEST DATE TO BE SEARCHED FOR (MO/DY/YR)
C       ENG  - ENGINE NUMBER TO BE SEARCHED FOR (10 CHARACTERS MAX. LEFT
C       JUSTIFIED)
C       HPFP - FUEL PUMP NUMBER TO BE SEARCHED FOR (10 CHARACTERS MAX. LEF
C       JUSTIFIED)
C       HPOP - OXIDIZER PUMP NUMBER TO BE SEARCHED FOR (10 CHARACTERS MAX.
C       LEFT JUSTIFIED)
C       ISTAT - SEARCH RESULTS
C        1-SEARCH SUCCESSFUL
C              0 - NONE FOUND
C             -1 - FILE READ ERROR
C
        SUBROUTINE STDIR(TEST,DATE,ENG,HPFP,HPOP,ISTAT)
        CHARACTER TEST*6,DATE*8,ENG*10,HPFP*10,HPOP*10
        CHARACTER TEST1*6,DATE1*8,ENG1*10,HPFP1*10,HPOP1*10
        CHARACTER TESTO*6,DATEO*8,ENGO*10,HPFPO*10,HPOPO*10,FD*4
        DATA TESTO,DATEO,ENGO,HPFPO,HPOPO /5*'          '/
        DATA FD/'TDIR'/
        IF(ISTAT.NE.0)GO TO 1
        LV=10
        OPEN(LV,FILE=FD,IOSTAT=ISTAT)
        IF (ISTAT.NE.0) GO TO 999
 1      CONTINUE
        TEST1='0'
 5      DATE1='0'
        ENG1='0'
        HPFP1='0'
 6      HPOP1='0'
 10     READ(10,11,END=200,ERR=300)TEST1,DATE1,
       1ENG1,HPFP1,HPOP1
 11     FORMAT(A6,1X,A8,2X,3(A10,2X))
 20     CONTINUE
        IF(TEST.NE.TESTO) GO TO 30
 25     CONTINUE
        GO TO 40
 30     CONTINUE
        IF(TEST.NE.TEST1) GO TO 10
 40     CONTINUE
        IF(DATE.NE.DATEO) GO TO 50
        GO TO 60
 50     CONTINUE
        IF(DATE.NE.DATE1) GO TO 10
 60     CONTINUE
        IF(ENG.NE.ENGO) GO TO 70
        GO TO 80
 70     CONTINUE
        IF(ENG.NE.ENG1) GO TO 10
 80     CONTINUE
        IF(HPFP.NE.HPFPO) GO TO 90
        GO TO 100
 90     CONTINUE
        IF(HPFP.NE.HPFP1) GO TO 10
 100    CONTINUE
        IF(HPOP.NE.HPOPO) GO TO 110
        GO TO 120
 110    CONTINUE
```

```
110      CONTINUE
         IF(HPOP.NE.HPOP1) GO TO 10
120      ISTAT=1
```



```

         TEST=TEST1
130      DATE=DATE1
         ENG=ENG1
         HPFP=HPFP1
         HPOP=HPOP1
         RETURN
200      CLOSE(10,IOSTAT=IERR)
         ISTAT=0
         RETURN
300      ISTAT=-1
         RETURN
999      WRITE (*,1000) ISTAT
1000     FORMAT (' FILE OPEN ERROR NO. ' I4)
         STOP
          END
```

Name	Type	Offset	P	Class
DATE	CHAR*8	4	*	
DATE0	CHAR*8	14214		
DATE1	CHAR*8	14266		
ENG	CHAR*10	8	*	
ENG0	CHAR*10	14222		
ENG1	CHAR*10	14274		
FD	CHAR*4	14252		
HPFP	CHAR*10	12	*	
HPFP0	CHAR*10	14232		
HPFP1	CHAR*10	14284		
HPOP	CHAR*10	16	*	
HPOP0	CHAR*10	14242		
HPOP1	CHAR*10	14294		
IERR	INTEGER*4	14328		
ISTAT	INTEGER*4	20	*	
LV	INTEGER*4	14256		
TEST	CHAR*6	0	*	
TEST0	CHAR*6	14208		
TEST1	CHAR*6	14260		

```
C SEARCH THE TIME HISTORY FILES TO FIND THE MAGNITUDE AT A GIVEN TIME
C USAGE: CALL STIMEH(TEST,CH,TIME,MAG,ISTAT)
C TEST-TEST NUMBER
C CH-CHANNEL NUMBER
C TIME-TIME FOR WHICH MAGNITUDE IS TO BE FOUND
C MAG-MAGNITUDE AT TIME
C ISTAT-CALL STATUS
C      -3=TIME EXCEEDS MAXIMUM TIME IN DATA FILE. MAXIMUM TIME AND
C    MAGNITUDE AT MAXIMUM IME ARE RETURNED IN TIME AND MAGNITUDE.
C      -2=TIME LESS THAN 0
C      -1=NORMAL RETURN
C      1-225=OPEN ERROR CODES
C
C  NOT VERIFIED ON IBM 12/85
C
       SUBROUTINE STIMEH(TEST,CH,TIME,MAG,ISTAT)
```

```
      CHARACTER TEST*6,CH*3
      CHARACTER FD*12,IFLT*1
      REAL MAG,MAGO,MAG1
```



```
C    **APPEARS TO BE MISSING FILTER DESCR
      WRITE(*,*)'ENTER IFLT, 1 CHAR, SUB STIMEH'
      READ(*,'(A1)')IFLT
C    **ADDED 12/85
C
      WRITE(FD,10)IFLT,TEST,CH
   10 FORMAT('T',A1,A6,'.',A3)
      LV=12
      IF(ISTAT.NE.0) RETURN
      OPEN(LV,FILE=FD,IOSTAT=ISTAT)
      IF(ISTAT.GT.0) RETURN
      ISTAT=-1
      READ(12,20,END=800,ERR=850) XTIMEO,XMAGO
 20   FORMAT(2G13.0)
      IF(XTIMEO.LT.TIME) GO TO 50
      ISTAT=-2
      GO TO 1000
   50 READ(12,20,END=800,ERR=850) XTIME1,XMAG1
      IF(XTIME1.GE.TIME) GO TO 70
      XTIMEO=XTIME1
      XMAGO=XMAG1
      GO TO 50
   70 MAG=XMAGO+(TIME-XTIMEO)*(XMAG1-XMAGO)/(XTIME1-XTIMEO)
      GO TO 1000
  800 MAG=XMAG1
      TIME=XTIME1
      ISTAT=-3
      GO TO 1000
  850 WRITE(*,875)FD
  875 FORMAT(' READ ERROR ON FILE',A12)
      CLOSE(12,IOSTAT=IERR)
      STOP
 1000 CLOSE(12,IOSTAT=IERR)
      RETURN
      END
```

Name	Type	Offset	P	Class
CH	CHAR*3	4	*	
FD	CHAR*12	14363		
IERR	INTEGER*4	14452		
IFLT	CHAR*1	14362		
ISTAT	INTEGER*4	16	*	
LV	INTEGER*4	14396		
MAG	REAL	12	*	
MAGO	REAL	*****		
MAG1	REAL	*****		
TEST	CHAR*6	0	*	
TIME	REAL	8	*	
XMAGO	REAL	14404		
XMAG1	REAL	14420		
XTIMEO	REAL	14400		
XTIME1	REAL	14416		

```
C$PROG RDTH
```

```
C USAGE: CALL RDTH(TEST,CH,IFLT,TIME,MAG,LEN,ISTAT)
C TEST-TEST NUMBER
C CH-CHANNEL NUMBER
```



```
C IFLT - FILTER CODE
C TIME - TIME VECTOR
C MAG-MAGNITUDE VECTOR
C LEN - DIMENSION F TIME AND MAG ON CALL NUMBER OF DATA POINTS ON RETURN
C ISTAT-CALL STATUS
C       0=NORMAL RETURN
C      -1=NUMBER OF DATA POINTS GREATER THEN LEN
C       1-225=OPEN ERROR CODES
        SUBROUTINE RDTH(TEST,CH,IFLT,TIME,MAG,LEN,ISTAT)
        CHARACTER TEST*6,CH*3
        CHARACTER FD*12
        REAL MAG(512),TIME(512)
        WRITE(FD,10)IFLT,TEST,CH
     10 FORMAT('T'A1,A6,'.',A3)
        LV=12
        IF(ISTAT.NE.0) RETURN
        OPEN(LV,FILE=FD,IOSTAT=ISTAT)
        IF(ISTAT.NE.0)RETURN
        LENM=LEN
        LEN=0
 15     LEN=LEN+1
        READ(12,20,END=1000,ERR=850) TIME(LEN),MAG(LEN)
 20     FORMAT(2G13.0)
        IF(LEN.LT.LENM)GO TO 15
        ISTAT=-1
        GO TO 1000
    850 WRITE(*,875)FD
    875 FORMAT(' READ ERROR ON FILE',A12)
        CLOSE(12,IOSTAT=IERR)
        STOP
   1000 CLOSE(12,IOSTAT=IERR)
        LEN=LEN-1
        RETURN
        END
```

Name	Type	Offset	P	Class
CH	CHAR*3	4	*	
FD	CHAR*12	14456		
IERR	INTEGER*4	14530		
IFLT	INTEGER*4	8	*	
ISTAT	INTEGER*4	24	*	
LEN	INTEGER*4	20	*	
LENM	INTEGER*4	14490		
LV	INTEGER*4	14486		
MAG	REAL	16	*	
TEST	CHAR*6	0	*	
TIME	REAL	12	*	

```
C$PROG ERROR
        SUBROUTINE ERROR(MESG,ISTAT)
        CHARACTER MESG*20
        WRITE(*,*)'SUBROUTINE ERROR'
        WRITE(*,1)MESG,ISTAT
 1      FORMAT(' ERROR 'A20,' ISTAT= 'I4)
```

```
      STOP
      END
```



```
Name     Type          Offset P Class

ISTAT    INTEGER*4          4 *
MESG     CHAR*20            0 *
```

```
    C
    C$PROG INTERP
    C
    C  INTERPOLATION ROUTINE
    C
          SUBROUTINE INTERP(X,Y,XX,YY,SLOPE,LEN)
          DIMENSION X(500),Y(500)
          IF(XX.GE.X(1)) GO TO 10
          YY=Y(1)
          RETURN
    10    XLST=X(1)
          YLST=Y(1)
          DO 20 J=2,LEN
1         IF(X(J).GT.XX)GO TO 30
1         XLST=X(J)
1   20    YLST=Y(J)
          YY=Y(LEN)
          RETURN
    30    SLOPE=(Y(J)-YLST)/(X(J)-XLST)
          YY=YLST+SLOPE*(XX-XLST)
          RETURN
          END
```

```
Name     Type          Offset P Class

J        INTEGER*4      14570
LEN      INTEGER*4         20 *
SLOPE    REAL              16 *
X        REAL               0 *
XLST     REAL           14562
XX       REAL               8 *
Y        REAL               4 *
YLST     REAL           14566
YY       REAL              12 *
```

```
    C$PROG RDPSD
    C      PURPOSE:  READ THE CONTENTS OF A PSD DATTA FILE AND PUT
    C                FREQ. IN X AND PUT MAG. IN Y AND PUT COMPOSITE
    C                IN COMP
    C
    C      TEST - TEST NUMBER (1 WORD  6 CHARACTERS)
    C      CH    - CHANNEL NO.
    C      ISLICE - SLICE CODE
    C      LEN     - DIMENSION OF X AND Y WHEN CALLED - NUMBER OF DATA
    C                POINT ON RETURN
    C      !!  Y MUST BE DIMENSIONED ATLEAST LEN=10 IN MAIN PROGRAM  !!
    C      Y(LEN) = LEN
    C      T(LEN+2) = COMPOSITE ON PLOTS
```

```
C       Y(LEN+3) = COMPOSITE COMPUTED FROM DIGITIZED DATA
C       Y(LEN+4) = COMPOSITE COMPUTED FROM CONVERTED DATA
C       Y(LEN+5) = NO. OF POINT NOT ASSIGNED DURING CONVERSION
```



```
C       Y(LEN+6) = TIME SLICE
C       Y(LEN+7) = MAX. FREQUENCY
C       Y(LEN+8) - Y(LEN+10) = SPARE
C       ISTAT  - CALL STATUS
C                   )0  OPEN ERROR CODES
C                  ,-500 C(-500-CLOSE ERROR CODES)
C                  -1  READ ERROR
C                  -2  DATA EXCEEDED DIMENSION
C                  LEN CHECK FAILED
        SUBROUTINE RDPSD(TEST,CH,ISLICE,X,Y,COMB,LEN,ISTAT)
        CHARACTER CH*3,TEST*6
        CHARACTER FD12*12
        DIMENSION X(500),Y(500)
        WRITE(FD12,10)ISLICE,TEST,CH
   10   FORMAT(A2,A6,'.'A3)
        OPEN(12,FILE=FD12,IOSTAT=ISTAT)
        WRITE(5,'(A12)')FD12
        WRITE(*,*)FD12
        IF(ISTAT.NE.0)RETURN
        LENM=LEN+10
        LEN=0
   20   LEN=LEN+1
        READ(12,30,END=50,ERR=40)Y(LEN)
        IF(LEN.LT.LENM)GO TO 20
   30   FORMAT(G13.6)
        ISTAT=-2
        GO TO 60
   40   ISTAT=-1
        GO TO 60
   50   LEN=LEN-11
        COMB=Y(LEN+2)
        STEP=Y(LEN+7)/LEN
        DO 55 J=1,LEN
1  55   X(J)=J*STEP
C ??     IF(LEN.NE.IFIX(Y(LEN+1)))ISTAT=-3
   60   CLOSE(12,IOSTAT=IERR)
        IF(IERR.NE.0)ISTAT=-IERR-500
        RETURN
        END
```

Name	Type	Offset	P	Class
CH	CHAR*3	4	*	
COMB	REAL	20	*	
FD12	CHAR*12	14578		
IERR	INTEGER*4	14628		
ISLICE	INTEGER*4	8	*	
ISTAT	INTEGER*4	28	*	
J	INTEGER*4	14620		
LEN	INTEGER*4	24	*	
LENM	INTEGER*4	14604		
STEP	REAL	14616		
TEST	CHAR*6	0	*	
X	REAL	12	*	
Y	REAL	16	*	

```
      CPROG SPSD1
      C     PURPOSE:  SEARCH THE TEMPORARY PSD DIRECTORY FOR A GIVEN TEST NO.
```



```
      C     AND CHENNEL NO. -- RETURN WITH TIME SLICE
      C     MAX FREQ. AND SLICE CODE
      C
            SUBROUTINE SPSD1(TEST,CH,SLICE,FREQM,ISLICE,ISTAT,THRUST)
            CHARACTER TEST*6,TEST1*6,ISLICE*2,CHX*3
            CHARACTER FD13*8,CH*3
            DATA FD13 /'DPSD.TMP'/
            IF(ISTAT.EQ.0) GOTO 820
            OPEN(13,FILE=FD13,IOSTAT=ISTAT)
            IF(ISTAT.NE.0) GO TO 1180
      820   READ(13,830,END=860,ERR=1000)TEST1,ISLICE,CHX,SLICE,
           1FREQM,THRUST
      830   FORMAT(A6,1X,A2,1X,A3,3(1X,G13.6))
            IF (TEST1.GT.TEST)GOTO 860
            IF(TEST1.NE.TEST)GO TO 820
            IF(CHX.NE.CH)GO TO 820
            RETURN
      860   ISTAT=-1
            CLOSE(13,IOSTAT=IERR)
            IF(IERR.NE.0)GO TO 2000
            RETURN
      1000  WRITE(*,1110)
      1110  FORMAT(' READ ERROR IN SPSD1')
            STOP
      1180  WRITE(*,1190)ISTAT
      1190  FORMAT(' OPEN ERROR NO. 'I3,' ON FILE DSPD.TMP')
            STOP
      2000  WRITE(*,2010)IERR
      2010  FORMAT(' CLOSE ERROR NO. 'I3'ON FILE DPSD.TMP')
            STOP
            END
```

Name	Type	Offset	P	Class
CH	CHAR*3	4	*	
CHX	CHAR*3	14646		
FD13	CHAR*8	14632		
FREQM	REAL	12	*	
IERR	INTEGER*4	14678		
ISLICE	CHAR*2	16	*	
ISTAT	INTEGER*4	20	*	
SLICE	REAL	8	*	
TEST	CHAR*6	0	*	
TEST1	CHAR*6	14640		
THRUST	REAL	24	*	

```
      C
      C     CREATE TEMP FILE FOR SELECTED CHANNEL NUMBER
      C     SEARCH DPSD FOR ALL ENTRIES OF MATCHING CHANNEL
      C     WRITE THEM TO DPSD.TMP
            SUBROUTINE DPSDTEMP(TEST,CH,ISTAT)
            CHARACTER TEST*6,TEST1*6,ISLICE*2,CHX*3
            CHARACTER FD13*4,CH*3
            DATA FD13 /'DPSD'/
            OPEN(13,FILE=FD13,IOSTAT=ISTAT)
```

```
      IF(ISTAT.NE.0) GO TO 1180
      ICHAN=0
  820 READ(13,830,END=860,ERR=1000)TEST1,ISLICE,CHX,SLICE,
```



```
     1FREQM,THRUST
  830 FORMAT(A6,1X,A2,1X,A3,3(1X,G13.6))
      IF(CHX.NE.CH)GO TO 820
      IF (ICHAN.EQ.0) OPEN(6,FILE='DPSD.TMP')
      ICHAN=1
      WRITE(*,*)TEST1,CHX,' ',THRUST
      WRITE(6,830)TEST1,ISLICE,CHX,SLICE,FREQM,THRUST
      GOTO 820
  860 CONTINUE
      IF (ICHAN.EQ.0) WRITE(*,*)'NO PSD FILES FOR SELECTED CHAN'
      CLOSE(13,IOSTAT=IERR)
      IF(IERR.NE.0)GO TO 2000
      CLOSE(6,IOSTAT=IERR)
      IF (IERR.NE.0) GOTO 2000
      RETURN
 1000 WRITE(*,1110)
 1110 FORMAT(' READ ERROR IN DPSD')
      STOP
 1180 WRITE(*,1190)ISTAT
 1190 FORMAT(' OPEN ERROR NO. 'I3,' ON FILE DSPD')
      STOP
 2000 WRITE(*,2010)IERR
 2010 FORMAT(' CLOSE ERROR NO. 'I3'ON FILE DPSD OR .TMP')
      STOP
      END
```

Name	Type	Offset	P	Class
CH	CHAR*3	4	*	
CHX	CHAR*3	14808		
FD13	CHAR*4	14792		
FREQM	REAL	14816		
ICHAN	INTEGER*4	14796		
IERR	INTEGER*4	14852		
ISLICE	CHAR*2	14806		
ISTAT	INTEGER*4	8	*	
SLICE	REAL	14812		
TEST	CHAR*6	0	*	
TEST1	CHAR*6	14800		
THRUST	REAL	14820		

Name	Type	Size	Class
DPSDTE			SUBROUTINE
DSORT			SUBROUTINE
ERROR			SUBROUTINE
INTERP			SUBROUTINE
MAIN			PROGRAM
PEAK			SUBROUTINE
RDPSD			SUBROUTINE
RDTH			SUBROUTINE
SCHN			SUBROUTINE

SPSD SUBROUTINE
SPSD1 SUBROUTINE
STDIR SUBROUTINE

STIMEH SUBROUTINE

Pass One No Errors Detected
674 Source Lines

Section IV

SUMMARY OF THE SSME DYNAMIC PRESSURE ENVIRONMENT AT RATED POWER LEVEL

The appendices to this report include spectra for all the SSME tests, dynamic pressure measurements, and operating power levels provided by MSFC. A number of these measurements represent SSME operation over a limited power level range. However, the great majority of measurements include operation at RPL. In addition, an extensive data base of related vibration measurements has been developed at MSFC, representing SSME operation at RPL.

Average spectra were therefore extracted from the data base, to summarize the dynamic pressure measurements obtained at RPL. The procedure applied is summarized as follows.

1. Applying the data base sort routine, the spectra representing engine operation at RPL (99% - 101% RPL) were identified for each measurement.

2. The spectra were plotted in a waterfall format to permit a qualitative evaluation of measurement trends and repeatability.

3. These spectrum families were reviewed in detail and, based on engineering judgement, clearly deviant spectra were deleted from subsequent analysis. (It should be noted that <u>no</u> spectra were deleted from the original data base.)

4. Finally, the edited collection of spectra for each measurement was ensemble averaged using the statistical analysis subroutine, to obtain a single spectrum

representing each measurement at RPL. These results are presented as figures 4-1 through 4-27.

The above described spectra should provide a valuable reference to summarize the SSME dynamic pressure environment under nominal operating conditions. In applying these plots as a "benchmark", it should be noted that the spectra represent measurements with widely varying available sample size. For example, the auxilliary lox inlet pressure spectrum, figure 4-23, represents a single observation (PSD), whereas the main combustion chamber hot gas inlet pressure spectrum, figure 4-8, is the average of 44 PSD's available in the data base at RPL. Therefore, considerably more variation is to be expected about the former 'average' spectrum than the latter. An interpolation scheme to estimate the pressure spectrum over a wide range of engine power levels is discussed in the next section.

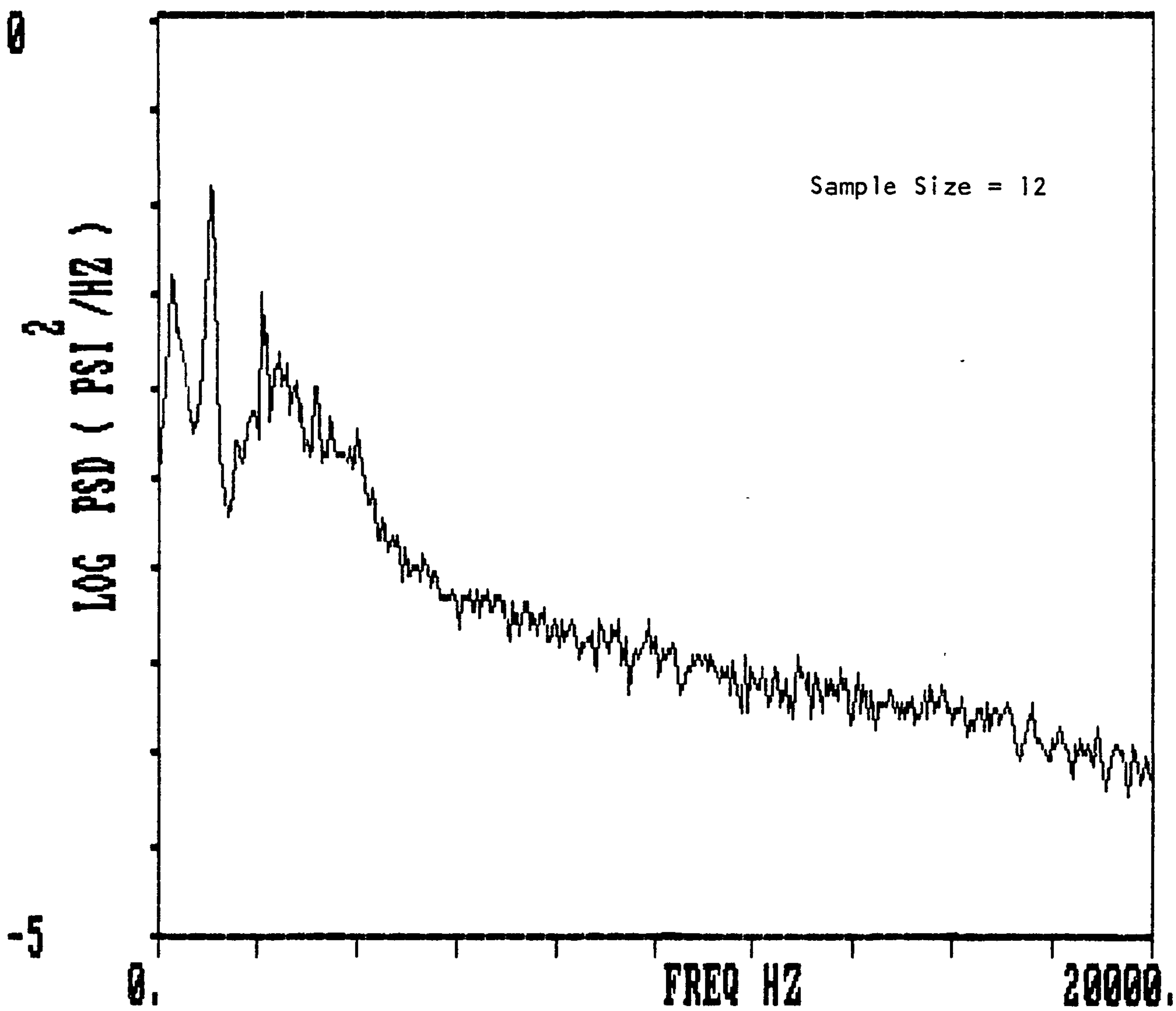


FIGURE 4-1. SPECTRUM OF LPFP IN PR AT RPL

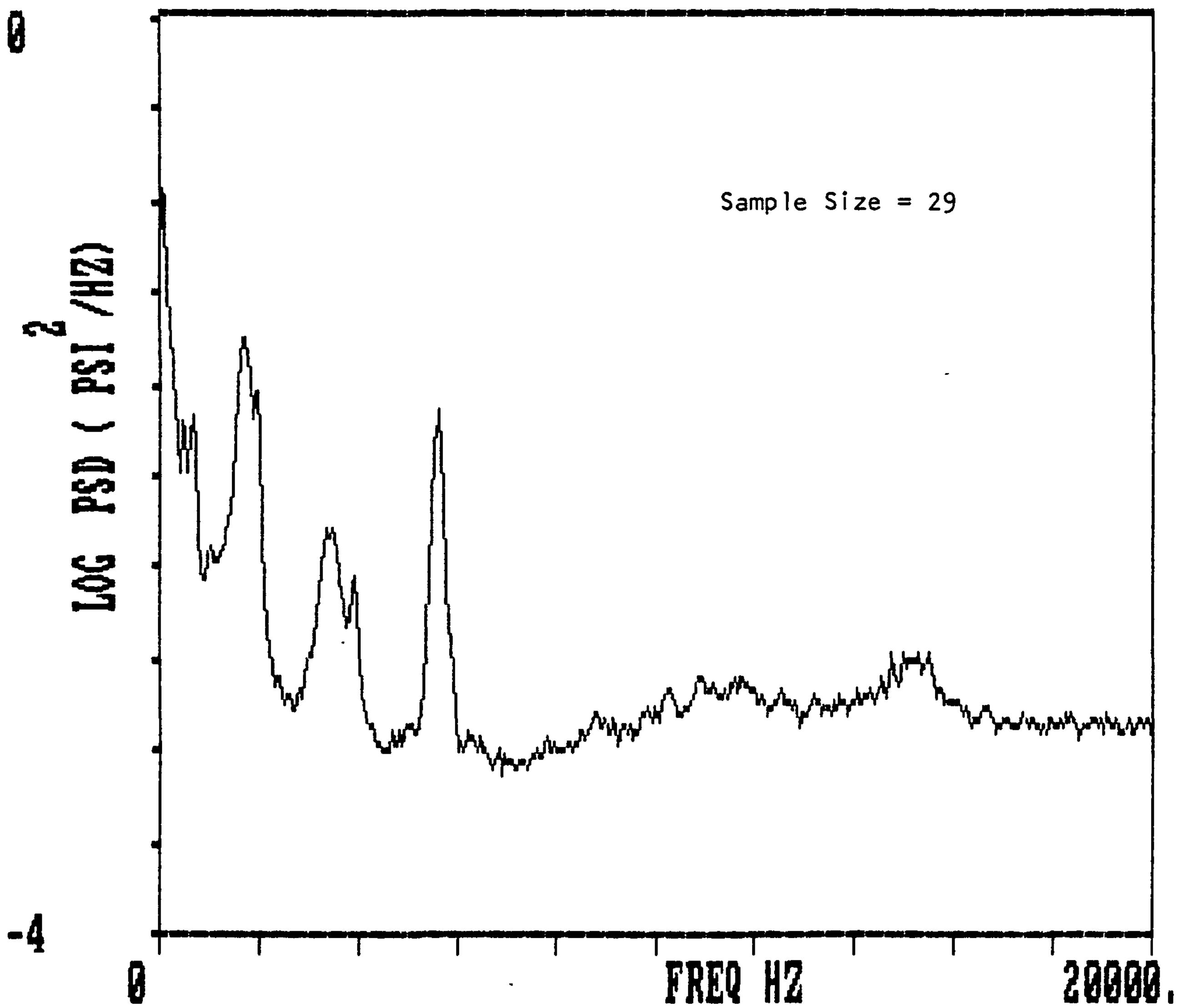


FIGURE 4-2. SPECTRUM OF LPOP DS PR AT RPL

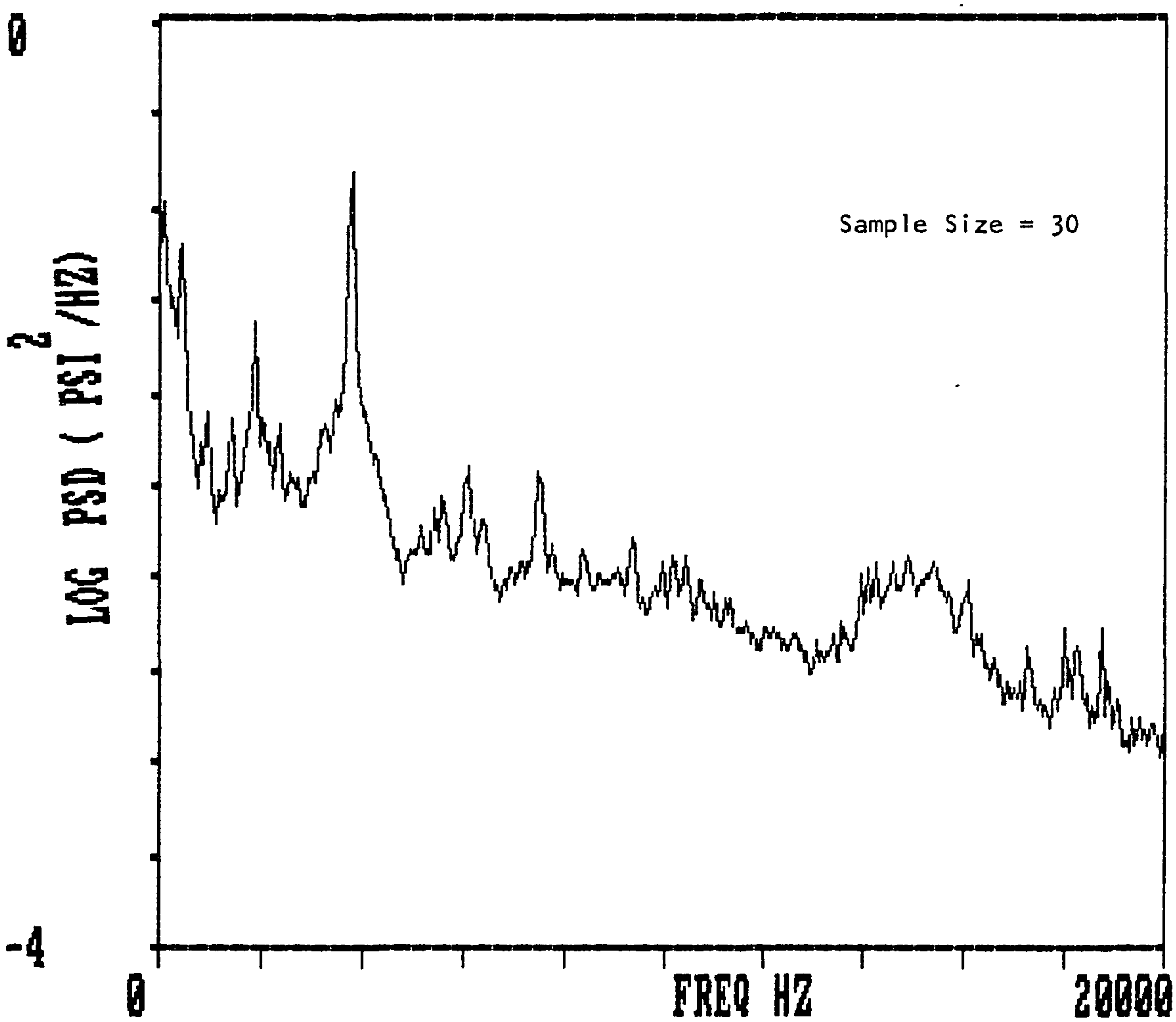


FIGURE 4-3. SPECTRUM OF HPOP BAL CAV PR2 AT RPL

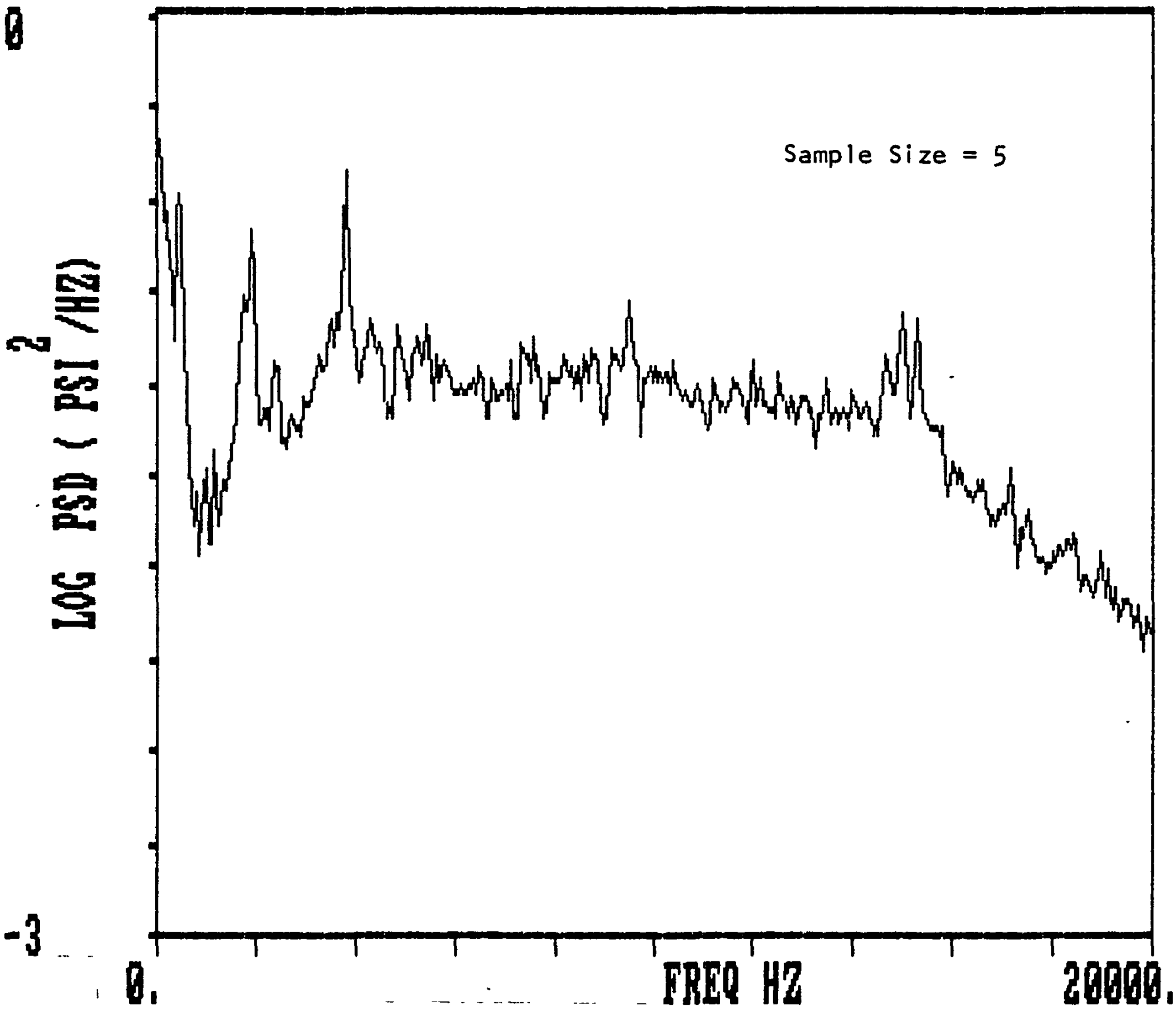


FIGURE 4-4. SPECTRUM OF HPOP BAL CAV PRI AT RPL

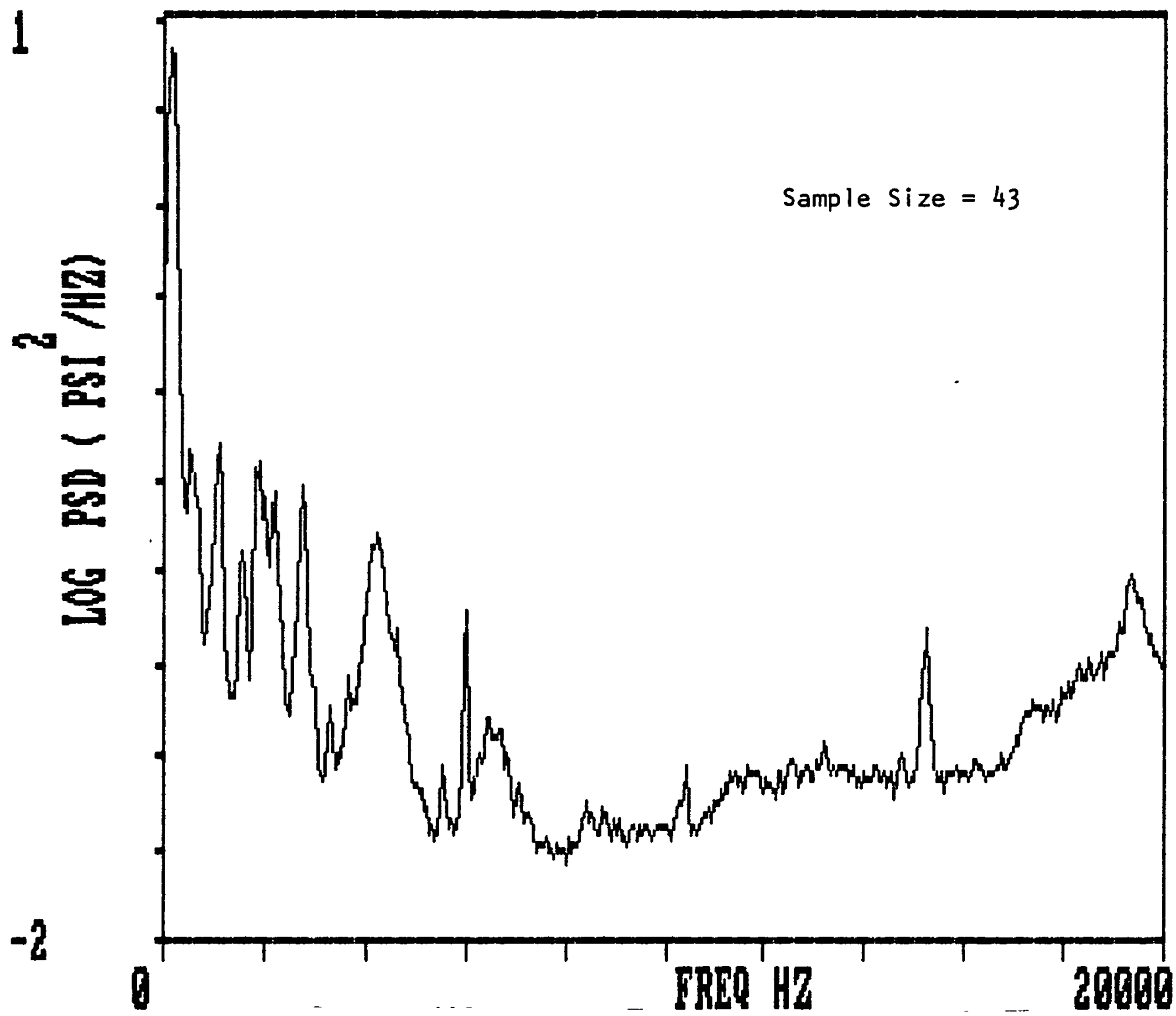


FIGURE 4-5. SPECTRUM OF HPOP DS PR AT RPL

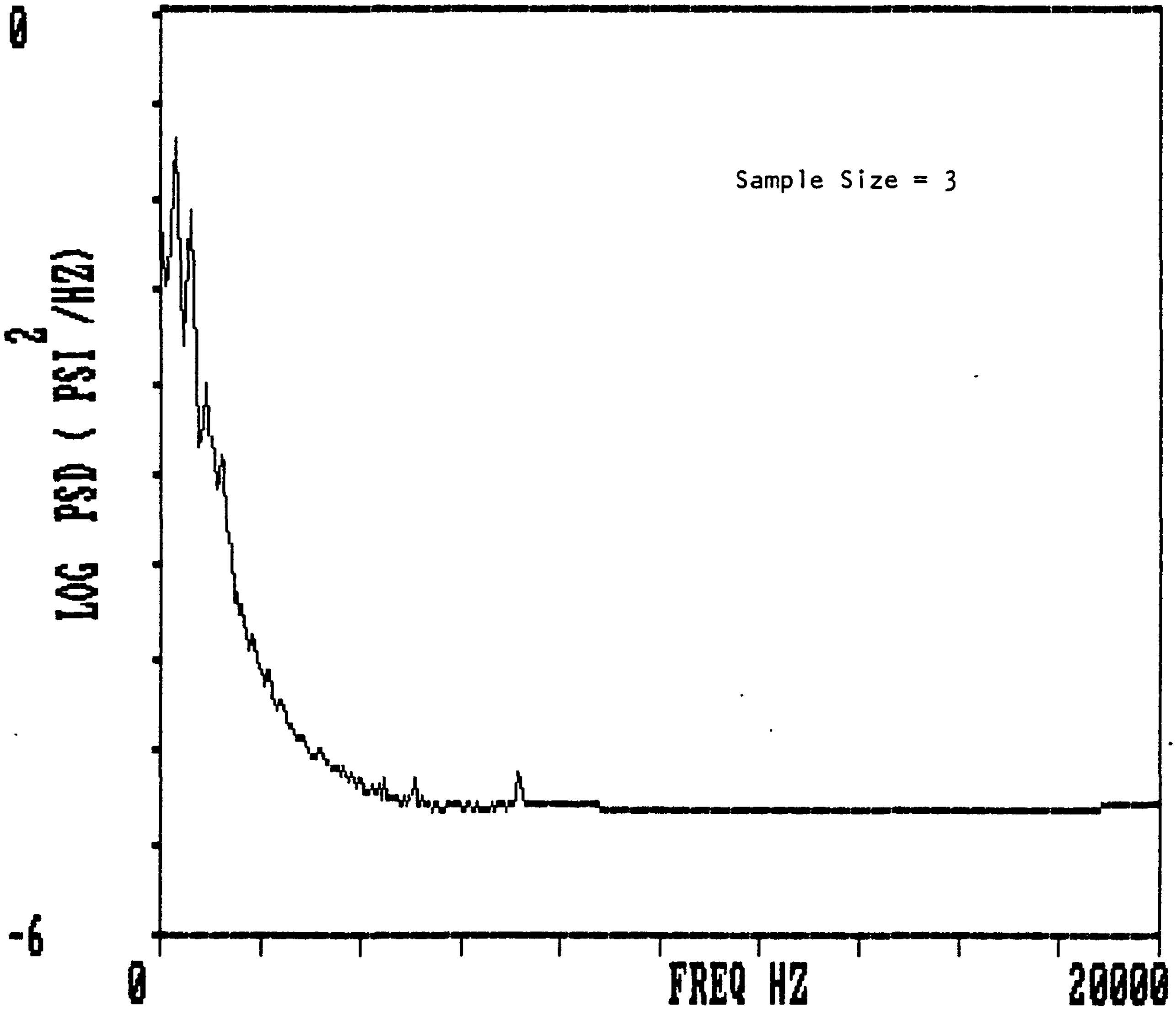


FIGURE 4-6. SPECTRUM OF LOX IN DUCT PR2 AT RPL

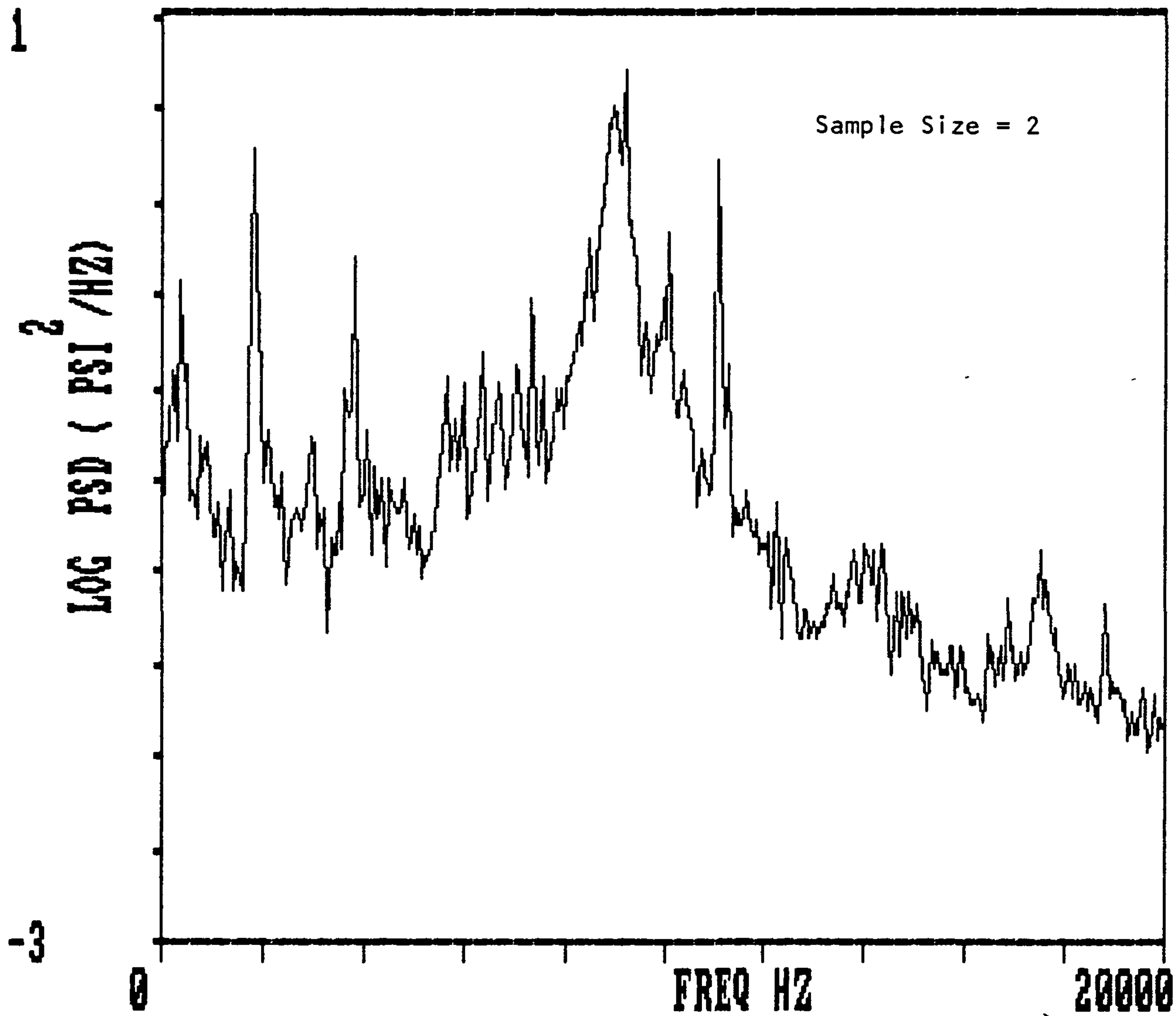


FIGURE 4-7. SPECTRUM OF HPOP DISC PR AT RPL

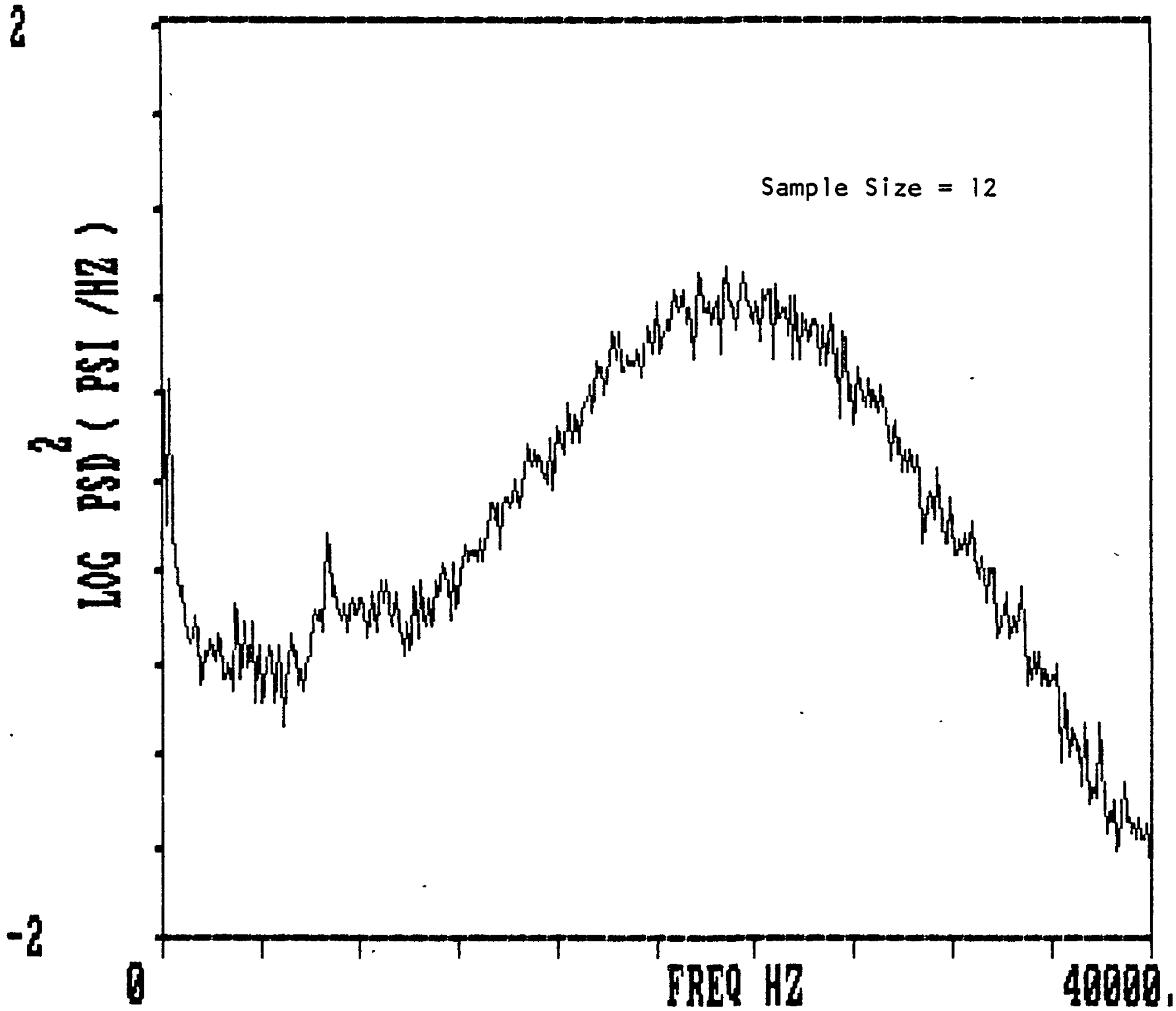


FIGURE 4-8. SPECTRUM OF MCC HOT GAS IN PR AT RPL

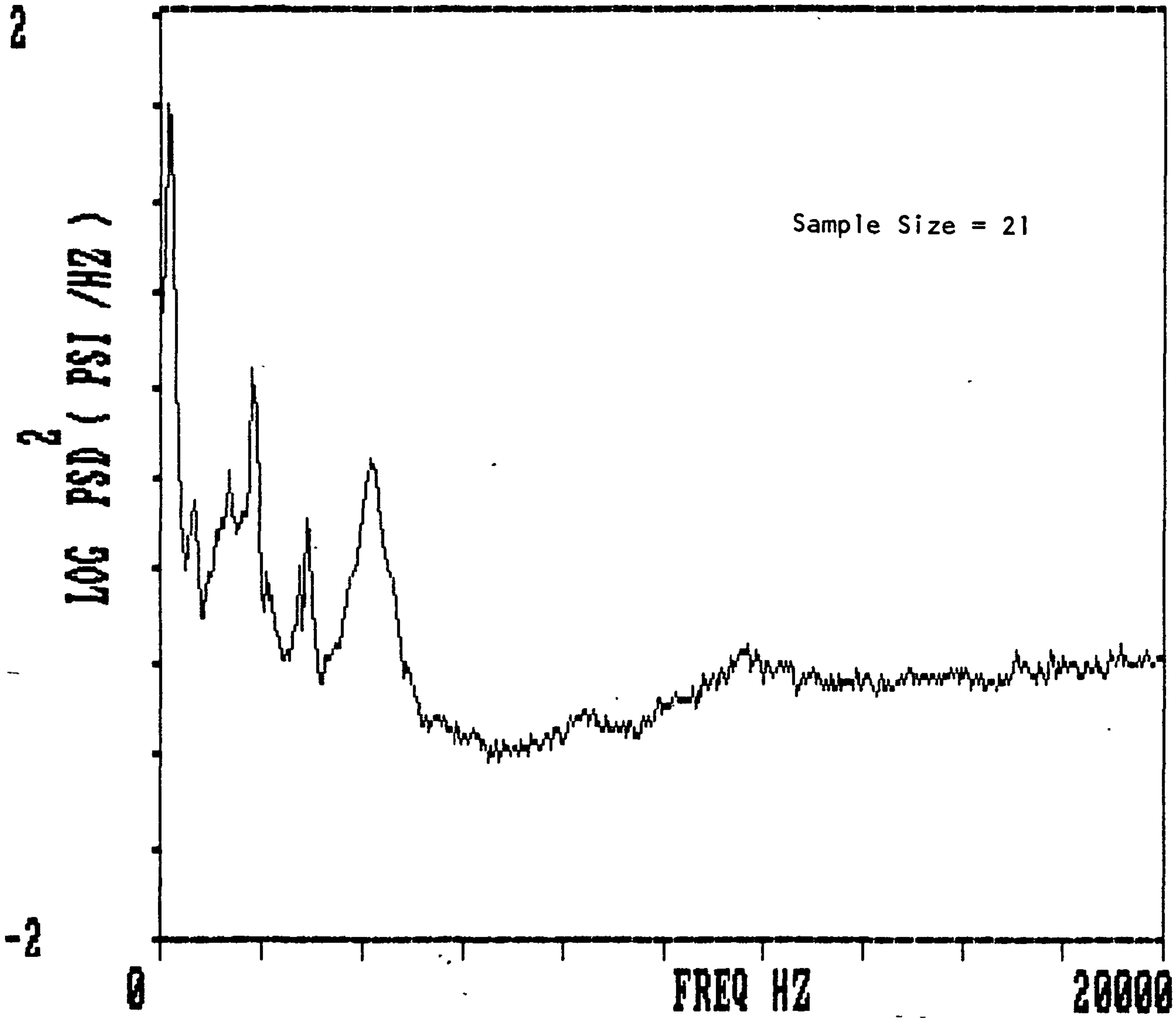


FIGURE 4-9. SPECTRUM OF PBP DS PR AT RPL

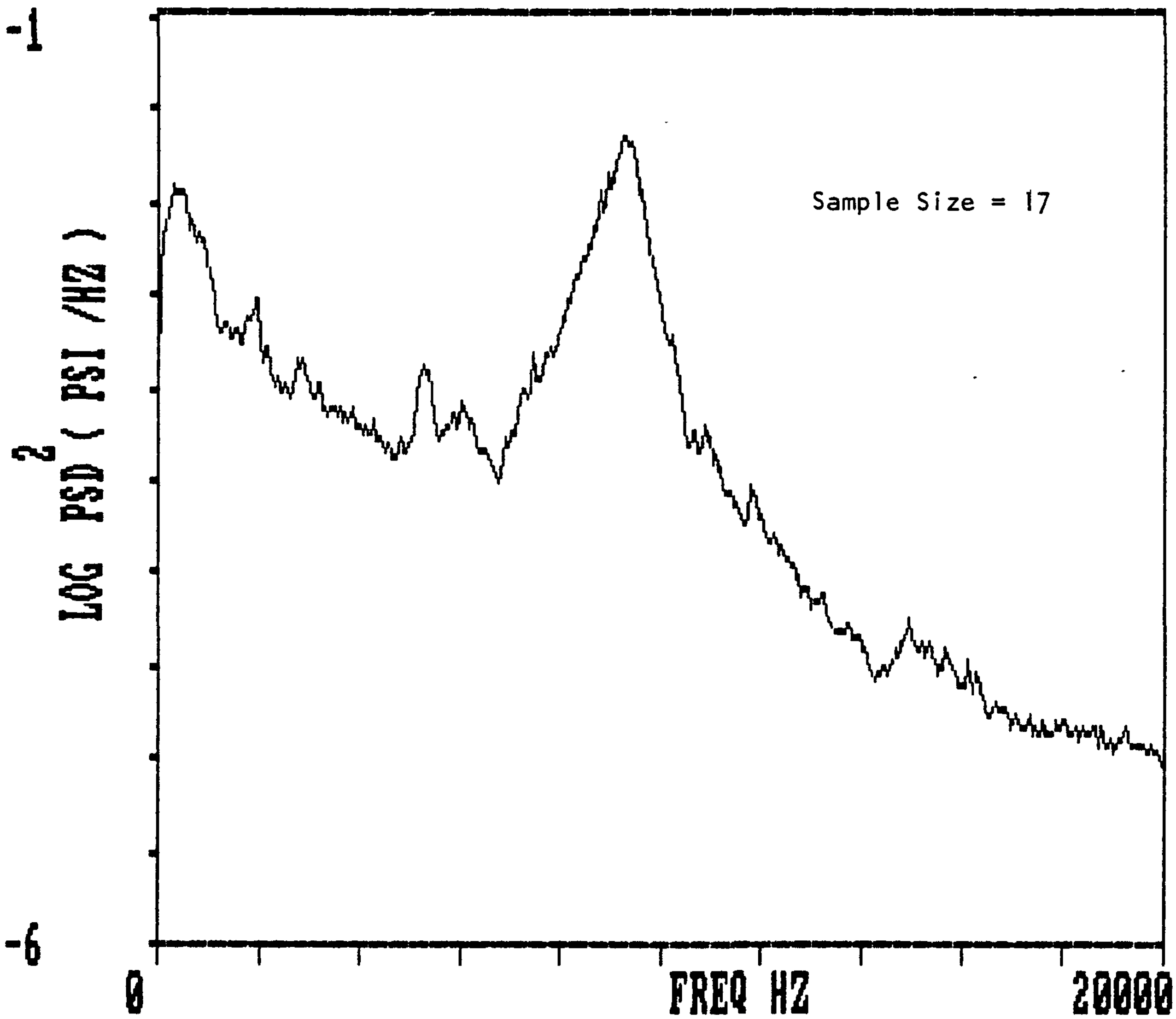


FIGURE 4-10. SPECTRUM OF LPFP DS PR AT RPL

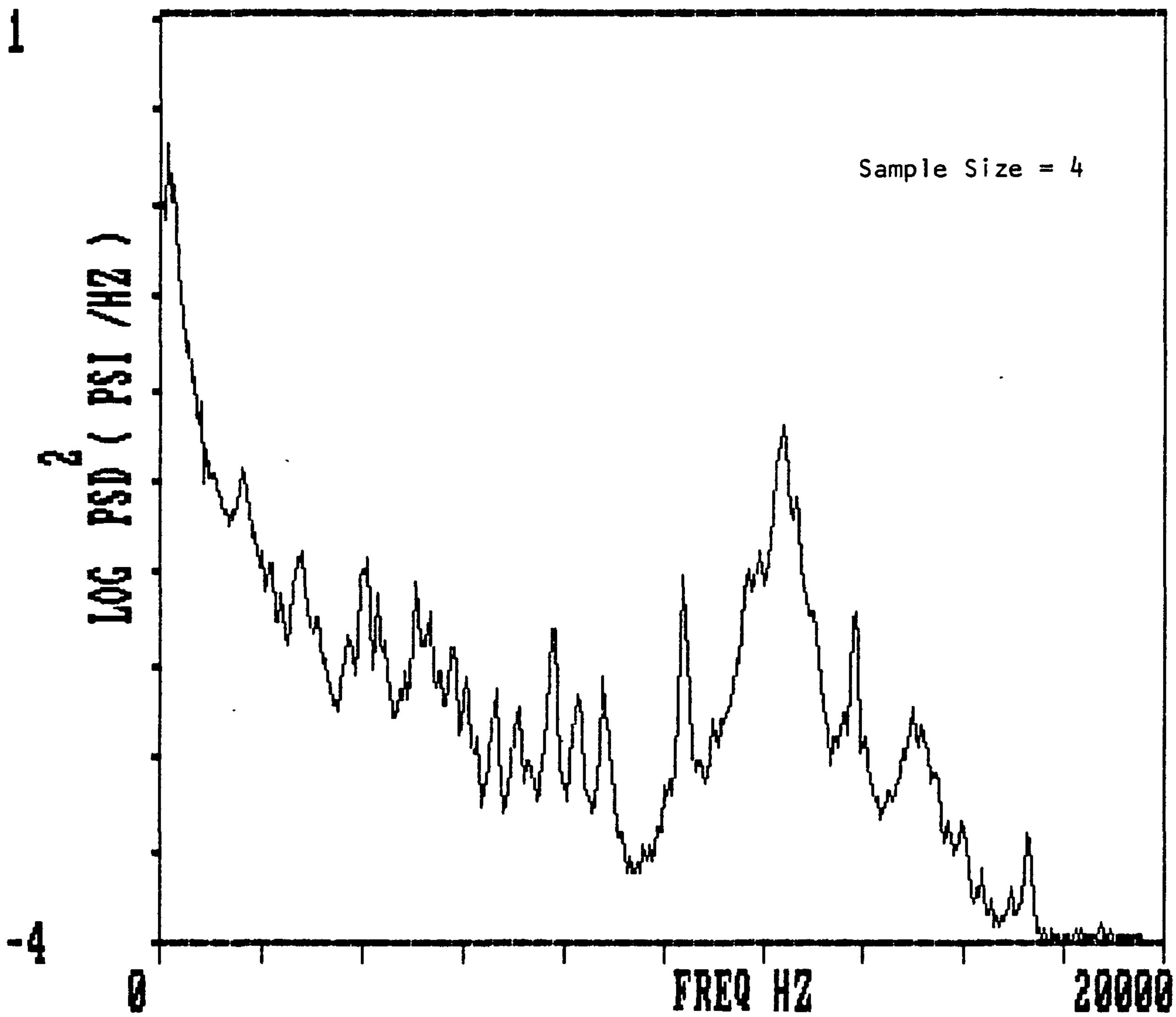


FIGURE 4-11. SPECTRUM OF FPB FUEL MAN PR AT RPL

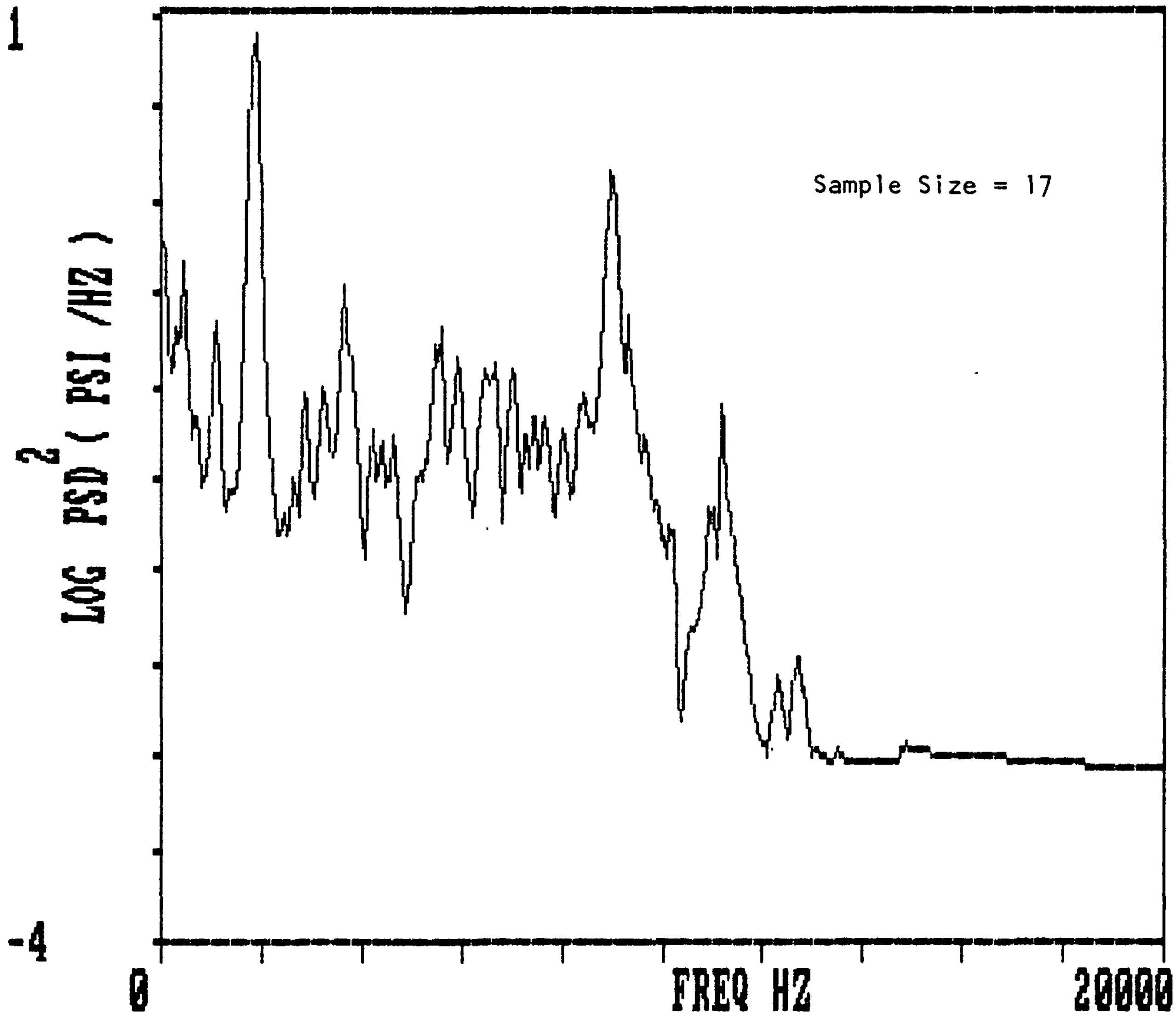


FIGURE 4-12. SPECTRUM LPOT TURB DR PR AT RPL

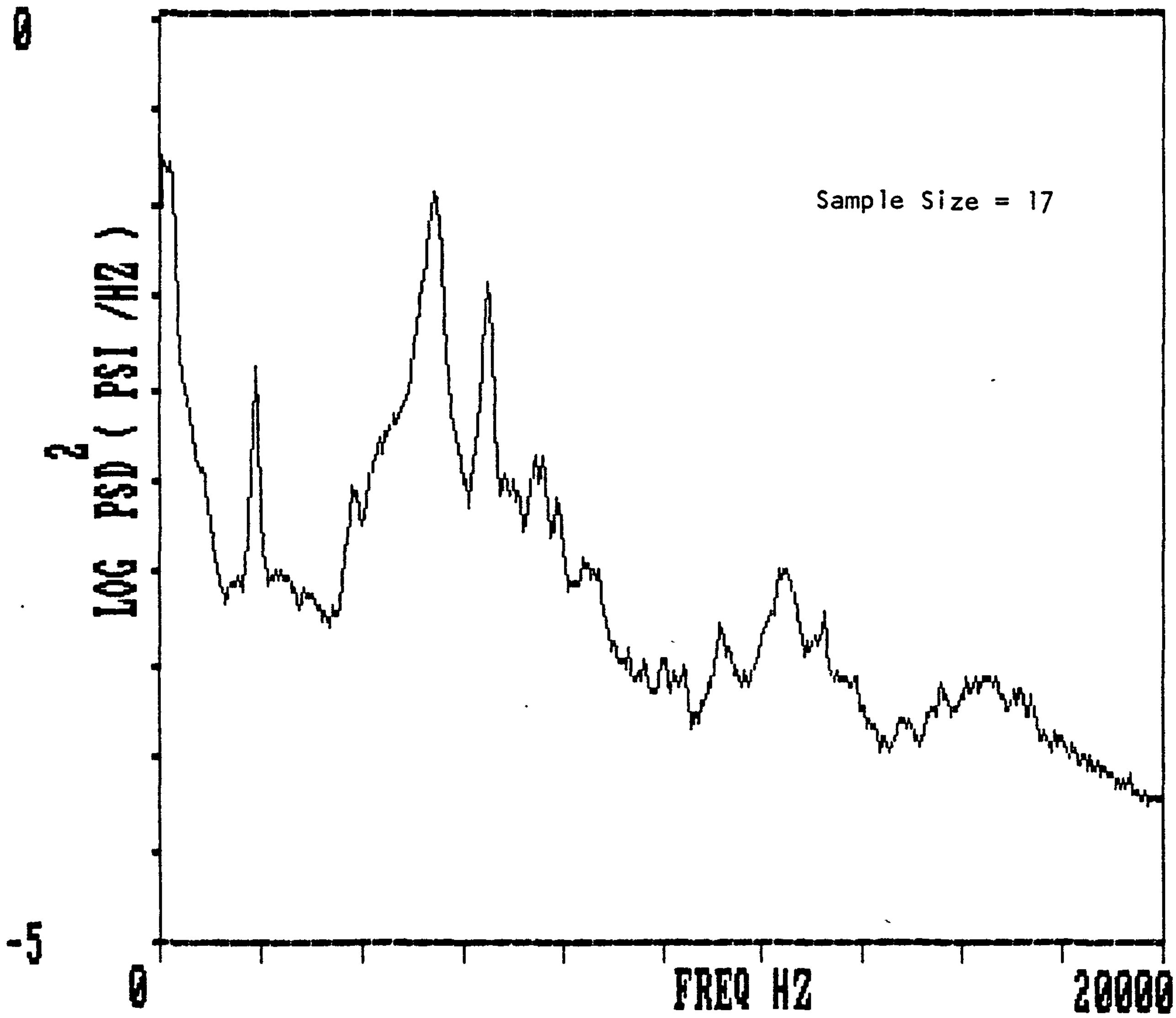


FIGURE 4-13. SPECTRUM OF OPB PC PR AT RPL

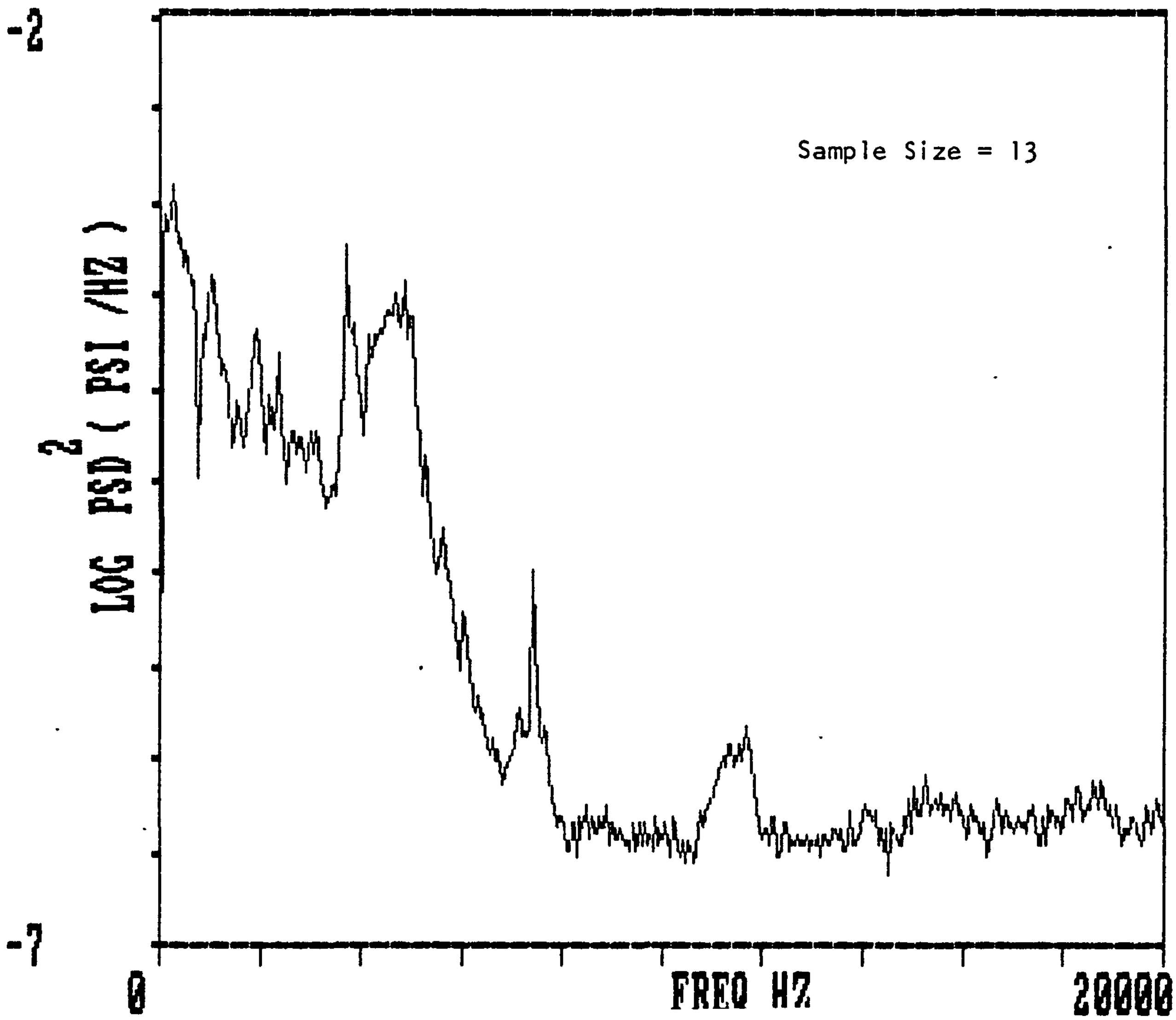


FIGURE 4-14. SPECTRUM OF HEX OUTLET AT RPL

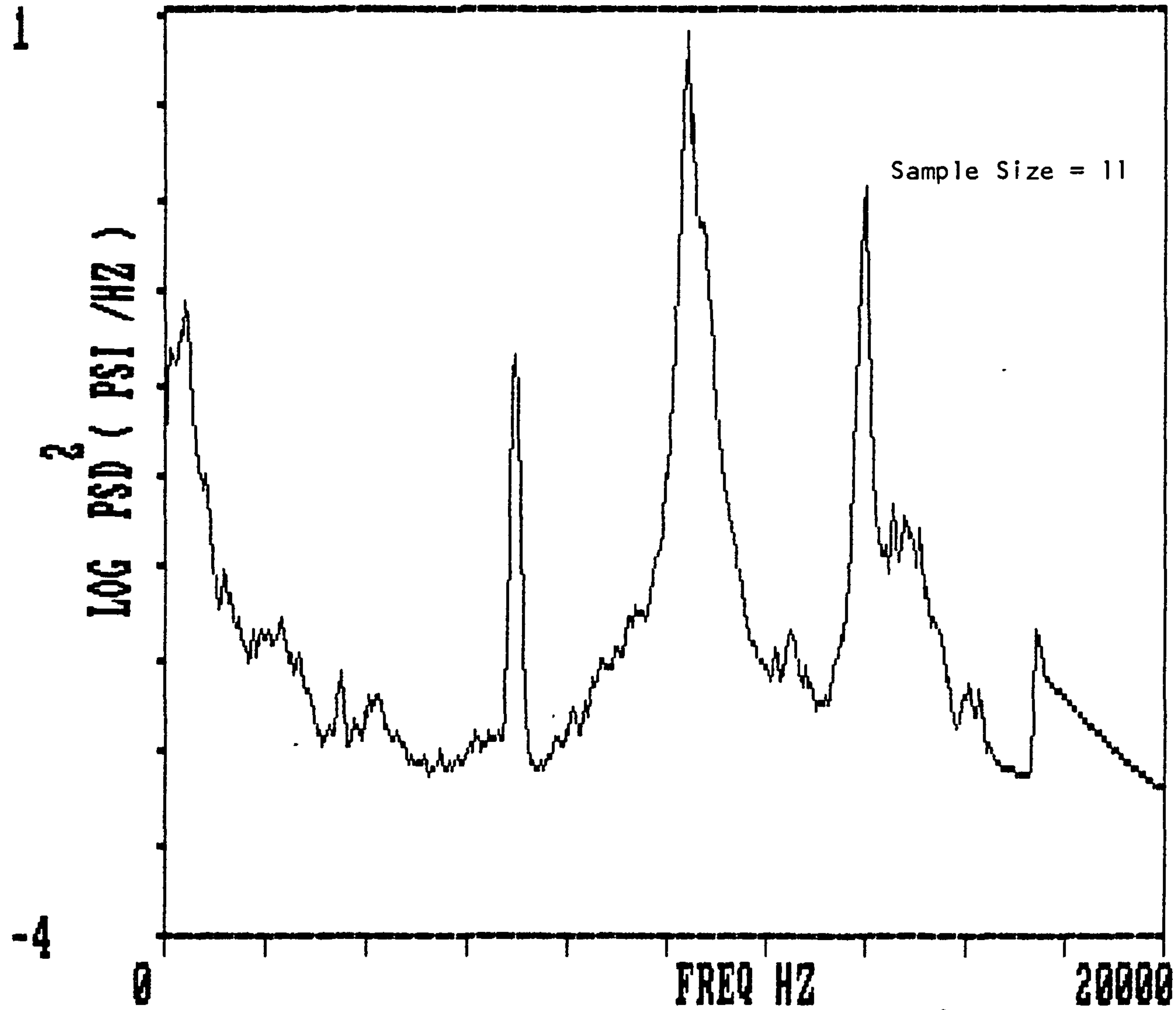


FIGURE 4-15. SPECTRUM OF HPFP DS PR AT RPL

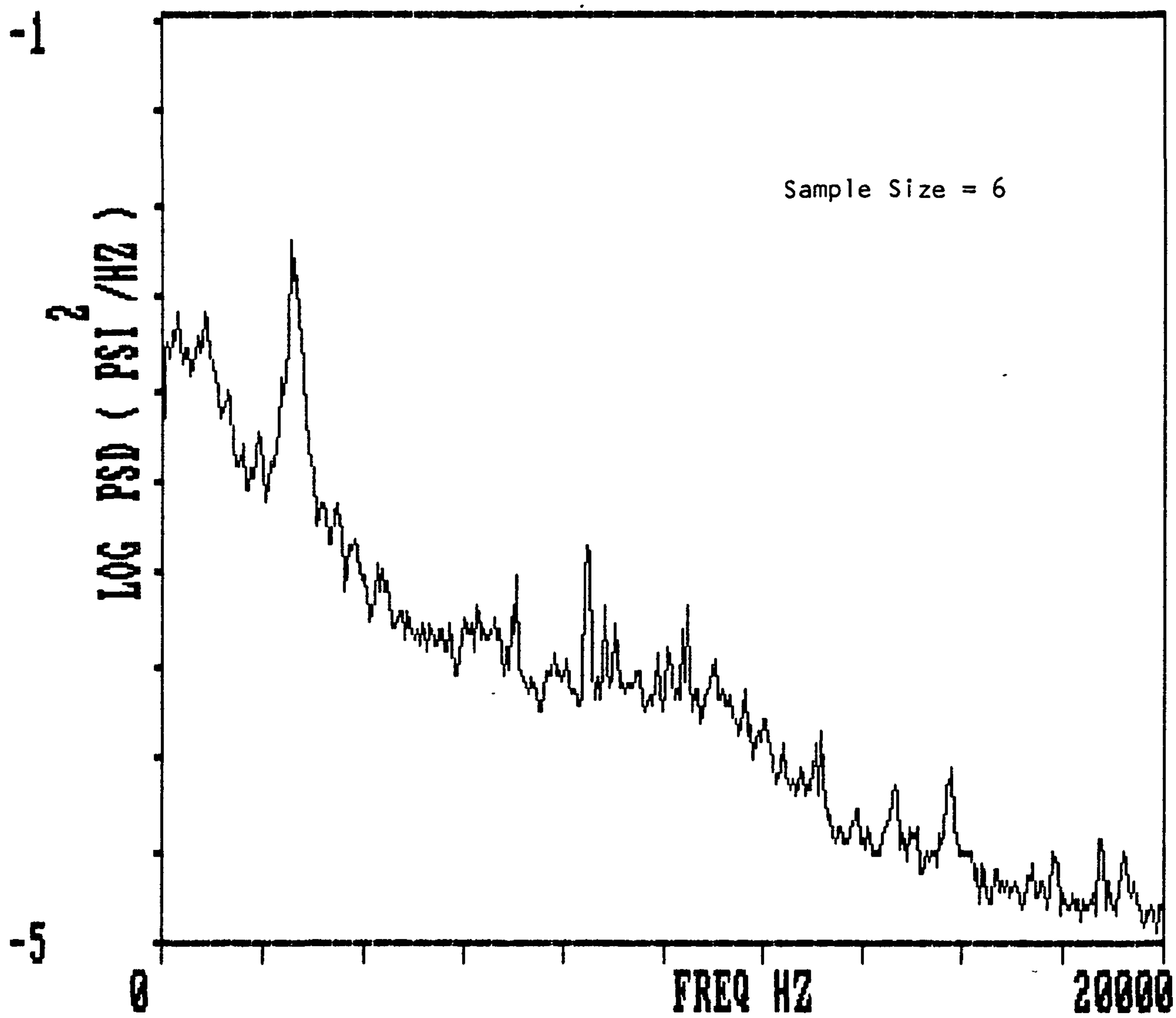


FIGURE 4-16. SPECTRUM OF HOT GAS MAN PR3 AT RPL

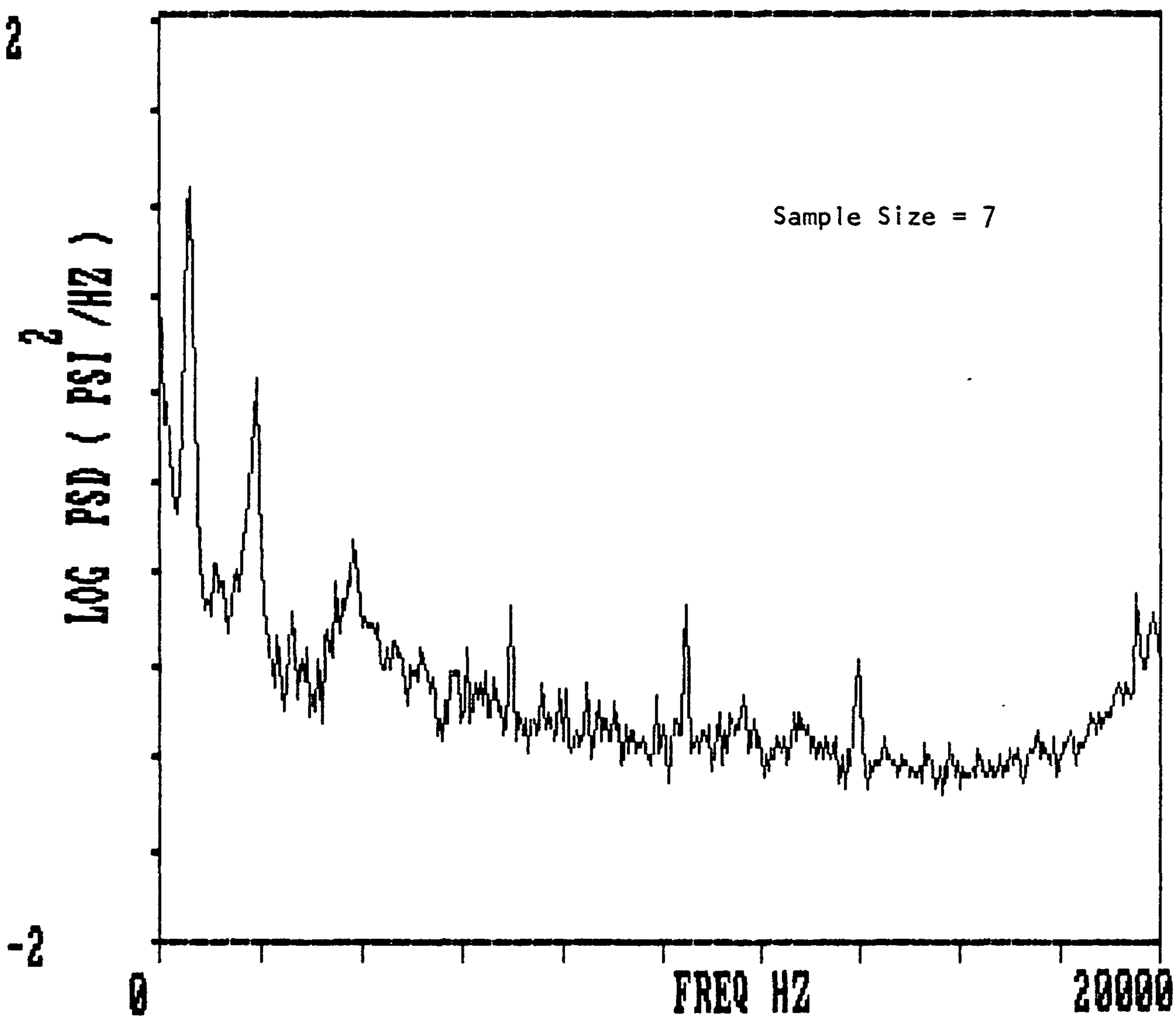


FIGURE 4-17. SPECTRUM OF HPFP BAL CAV PR AT RPL

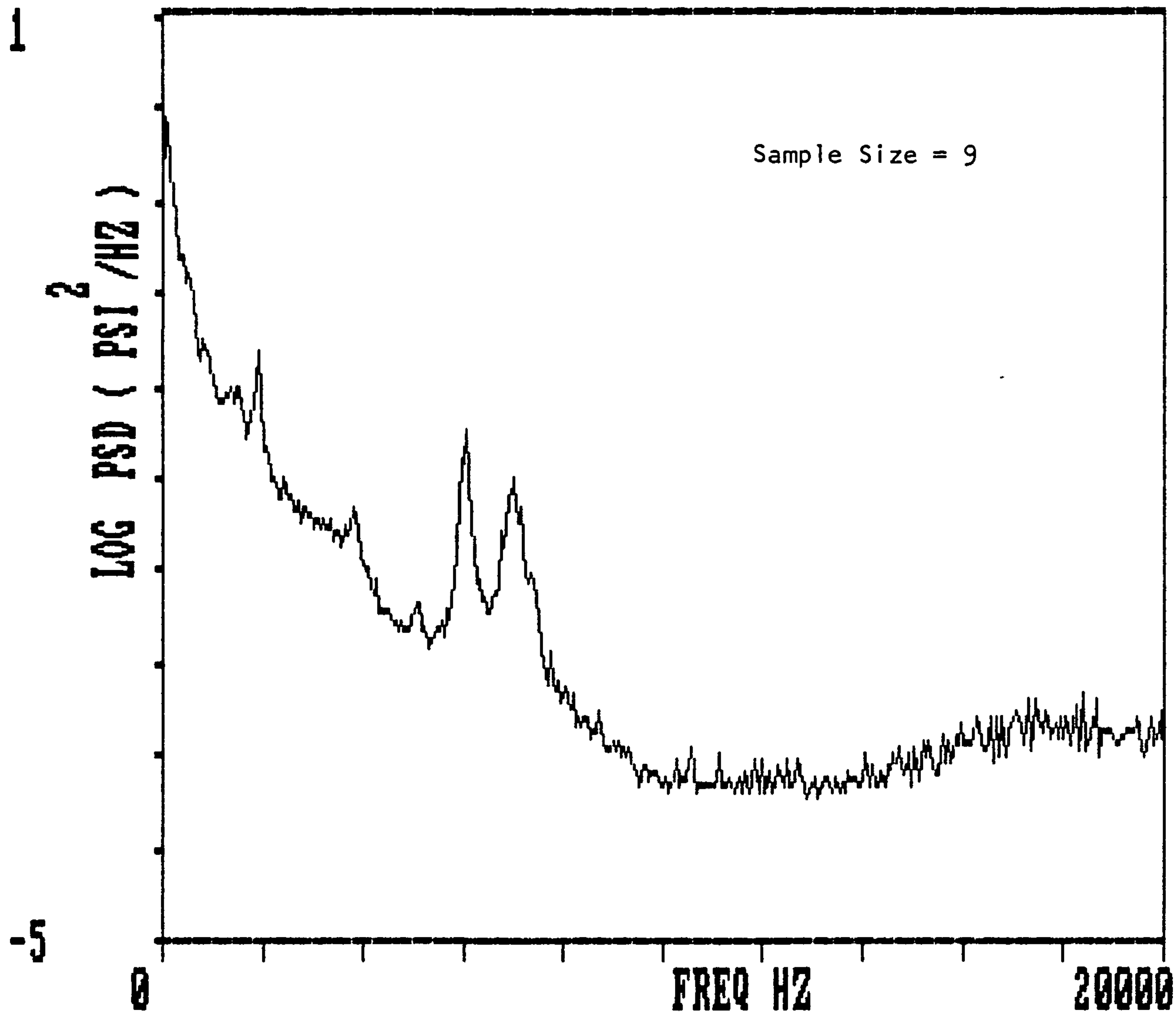


FIGURE 4-18. SPECTRUM OF MCC FUEL INJ PR AT RPL

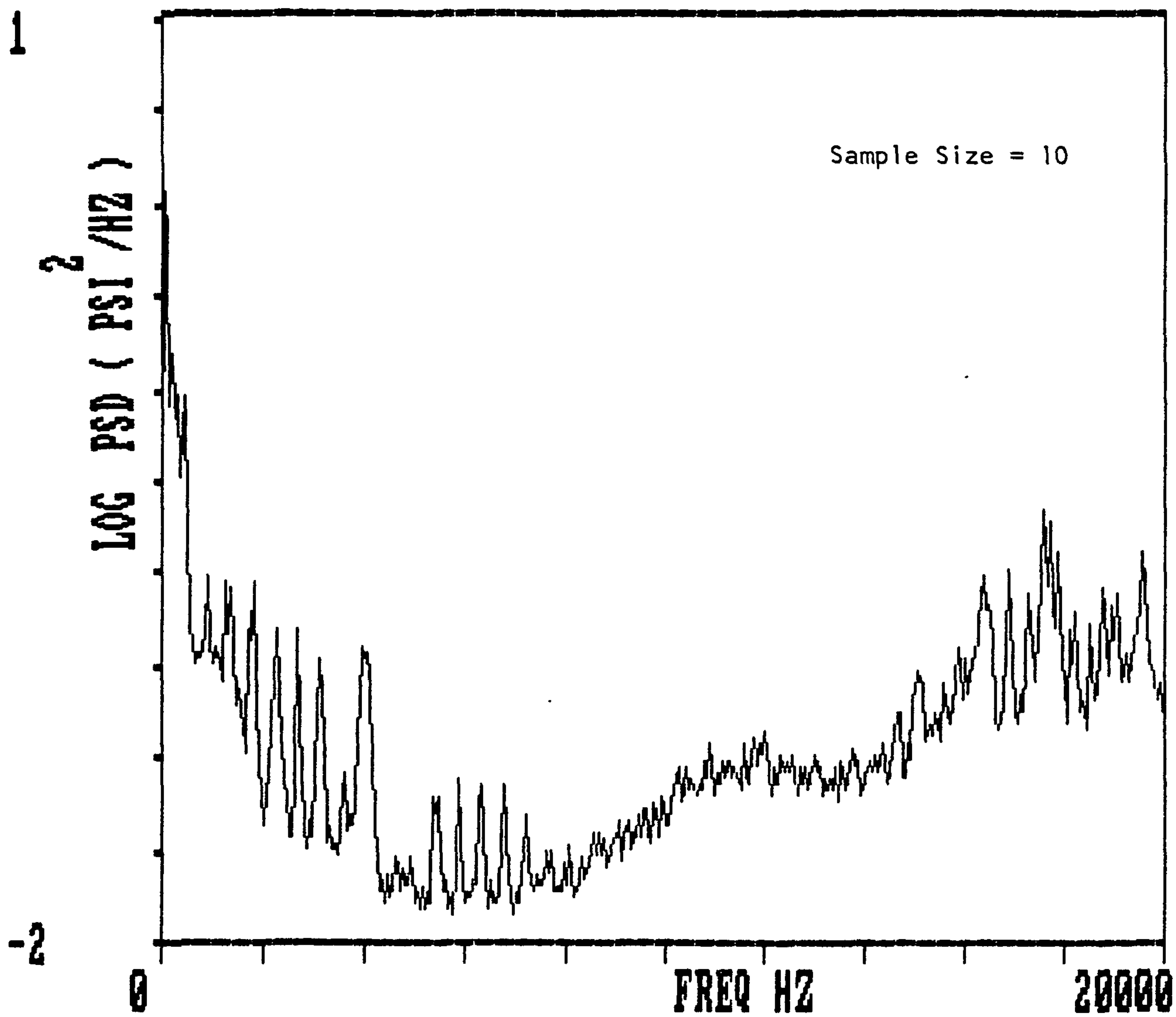


FIGURE 4-19. SPECTRUM OF FPB PC AT RPL

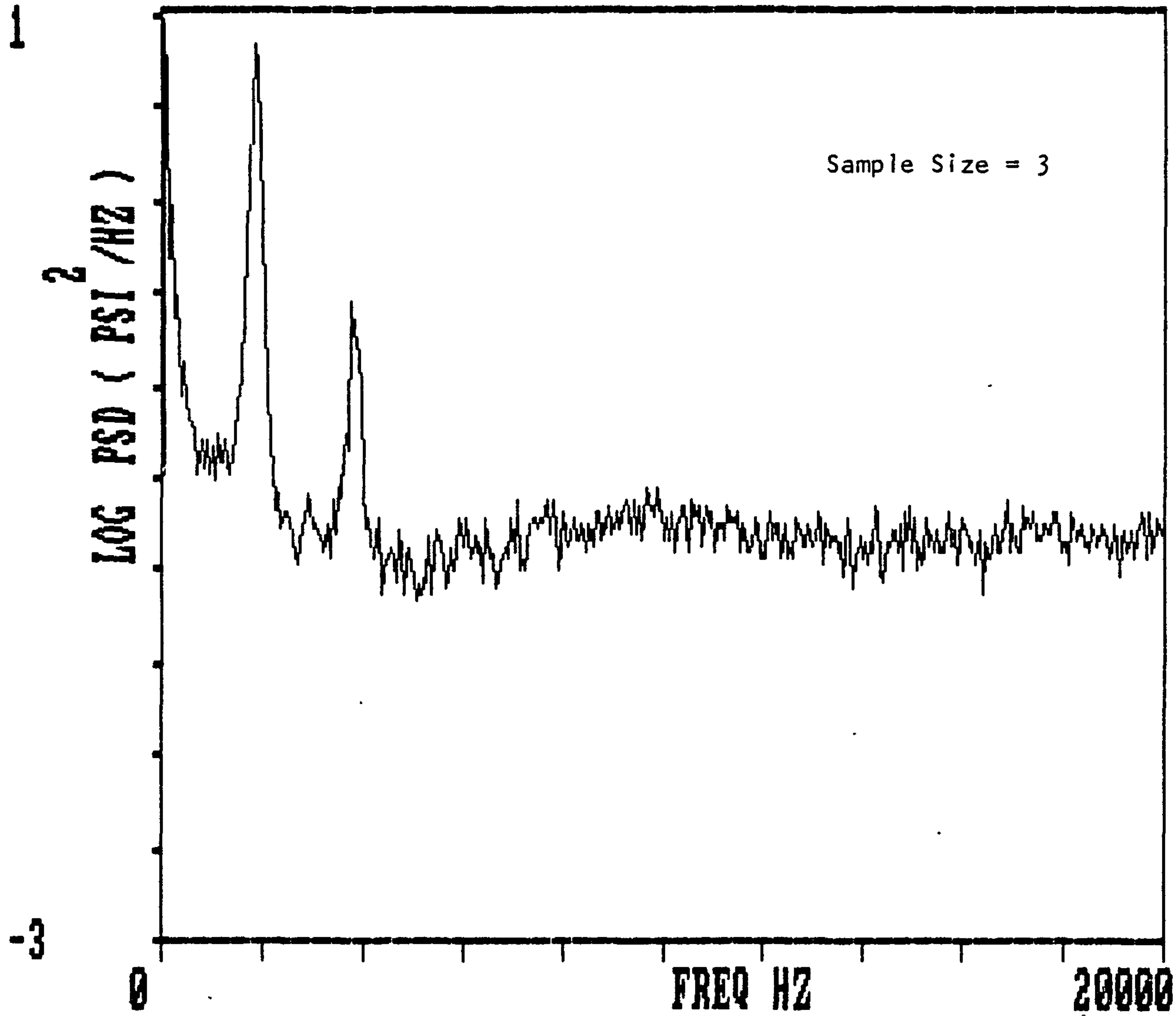


FIGURE 4-20. SPECTRUM OF MCC FUEL INJ PR DC AT RPL

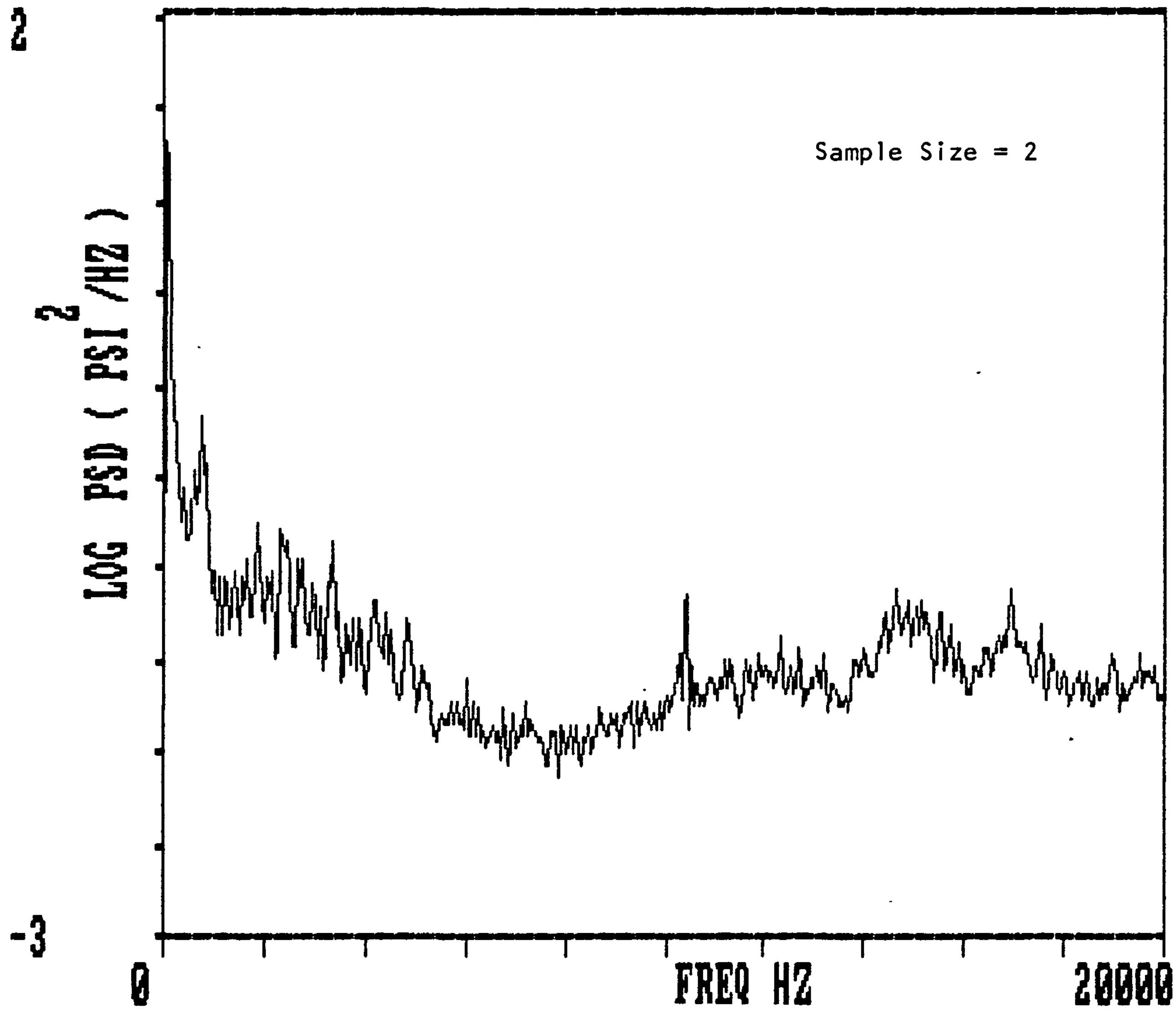


FIGURE 4-21. SPECTRUM OF OPB PL AT RPL

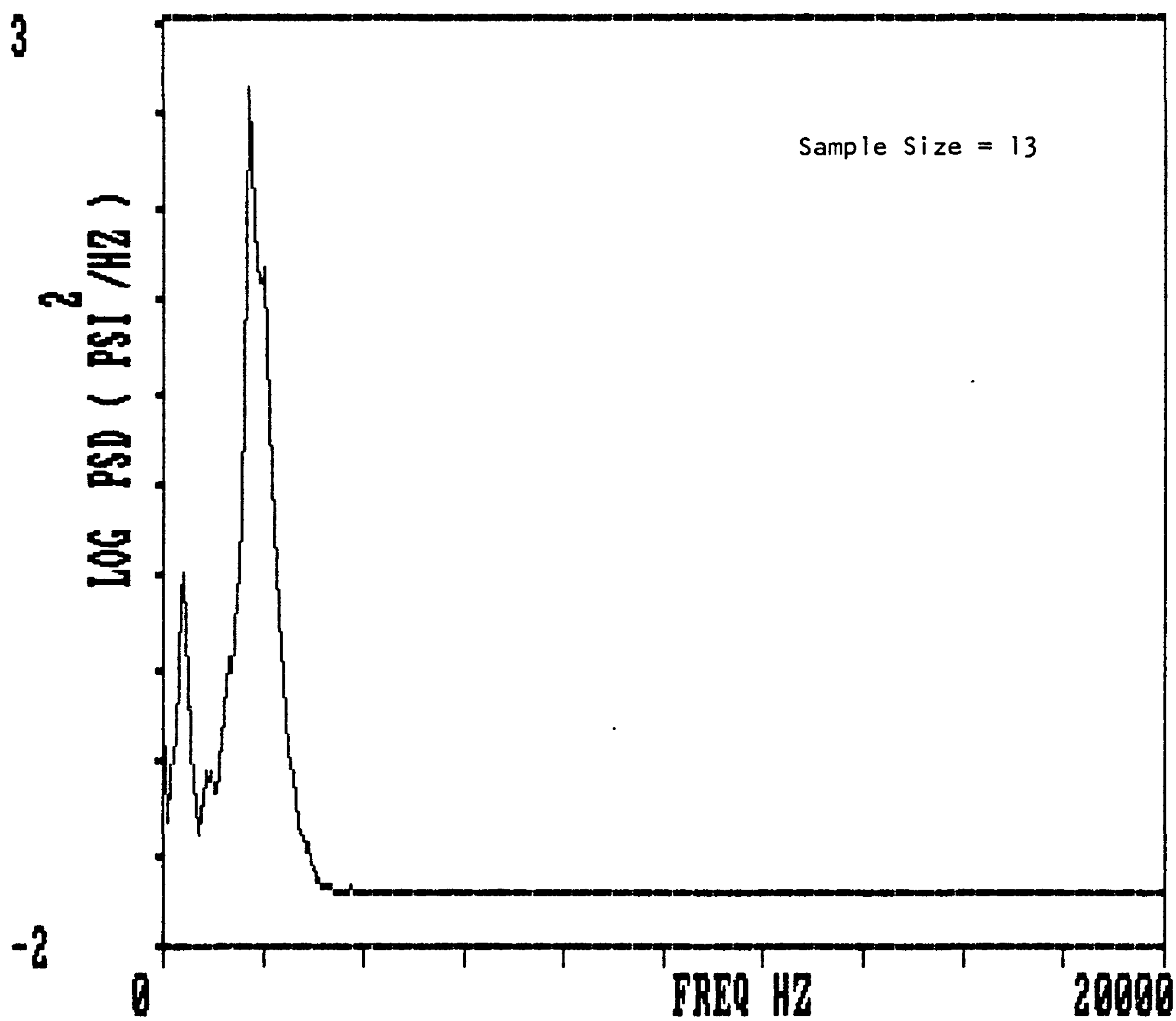


FIGURE 4-22. SPECTRUM OF HPOP IN PR AT RPL

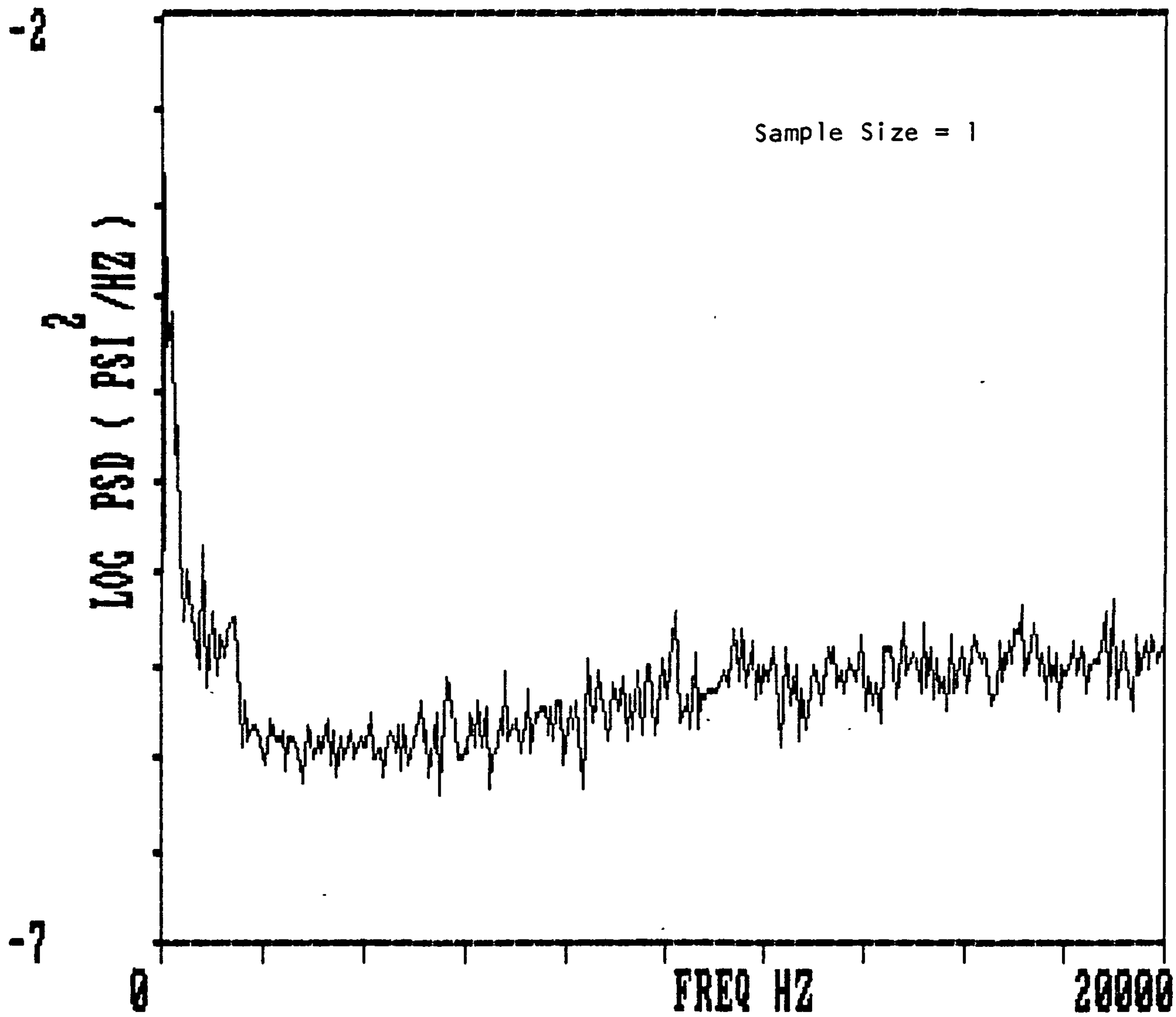


FIGURE 4-23. SPECTRUM OF AUX LX I PR AT RPL

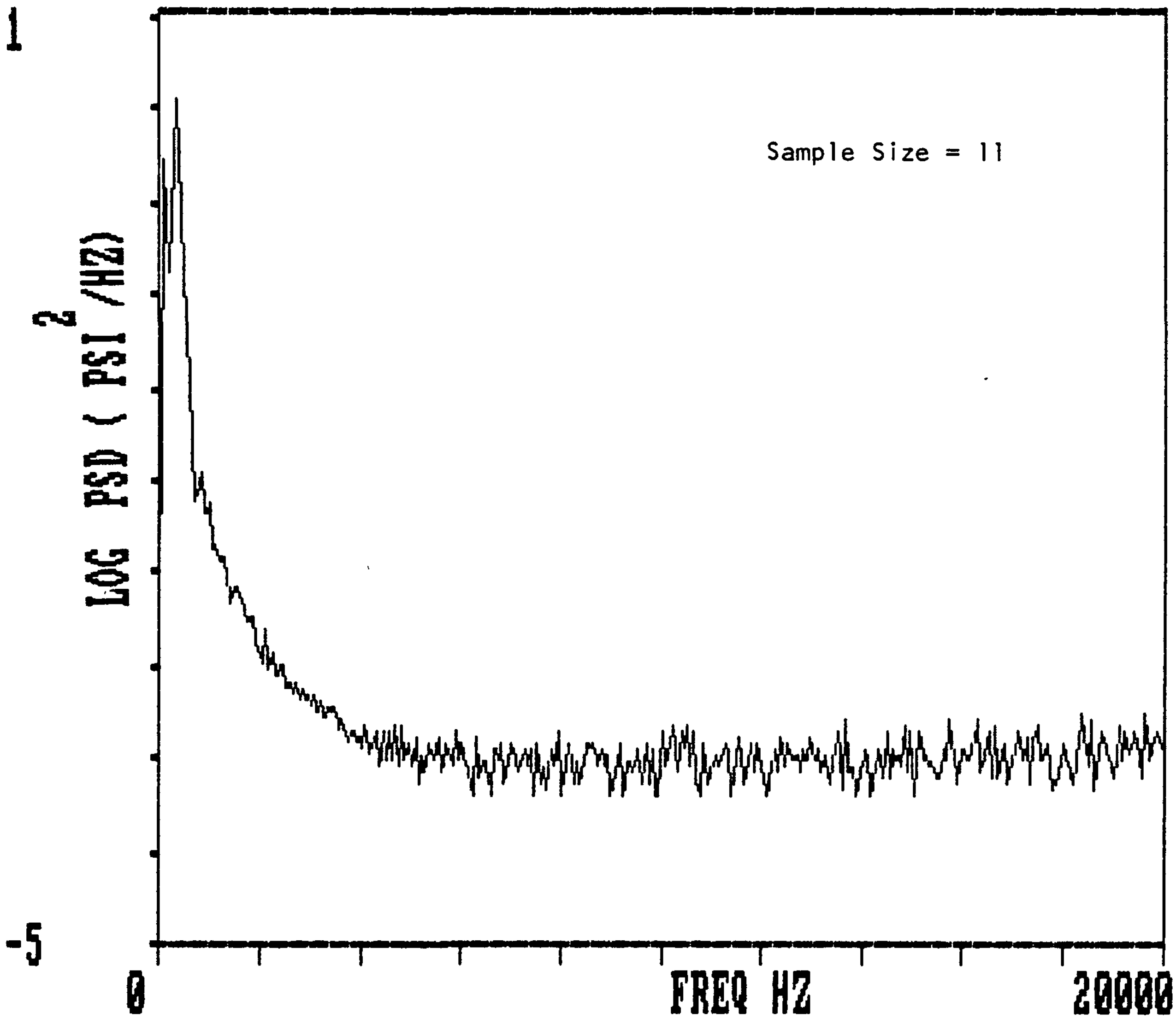


FIGURE 4-24. SPECTRUM OF LX I PR AT RPL

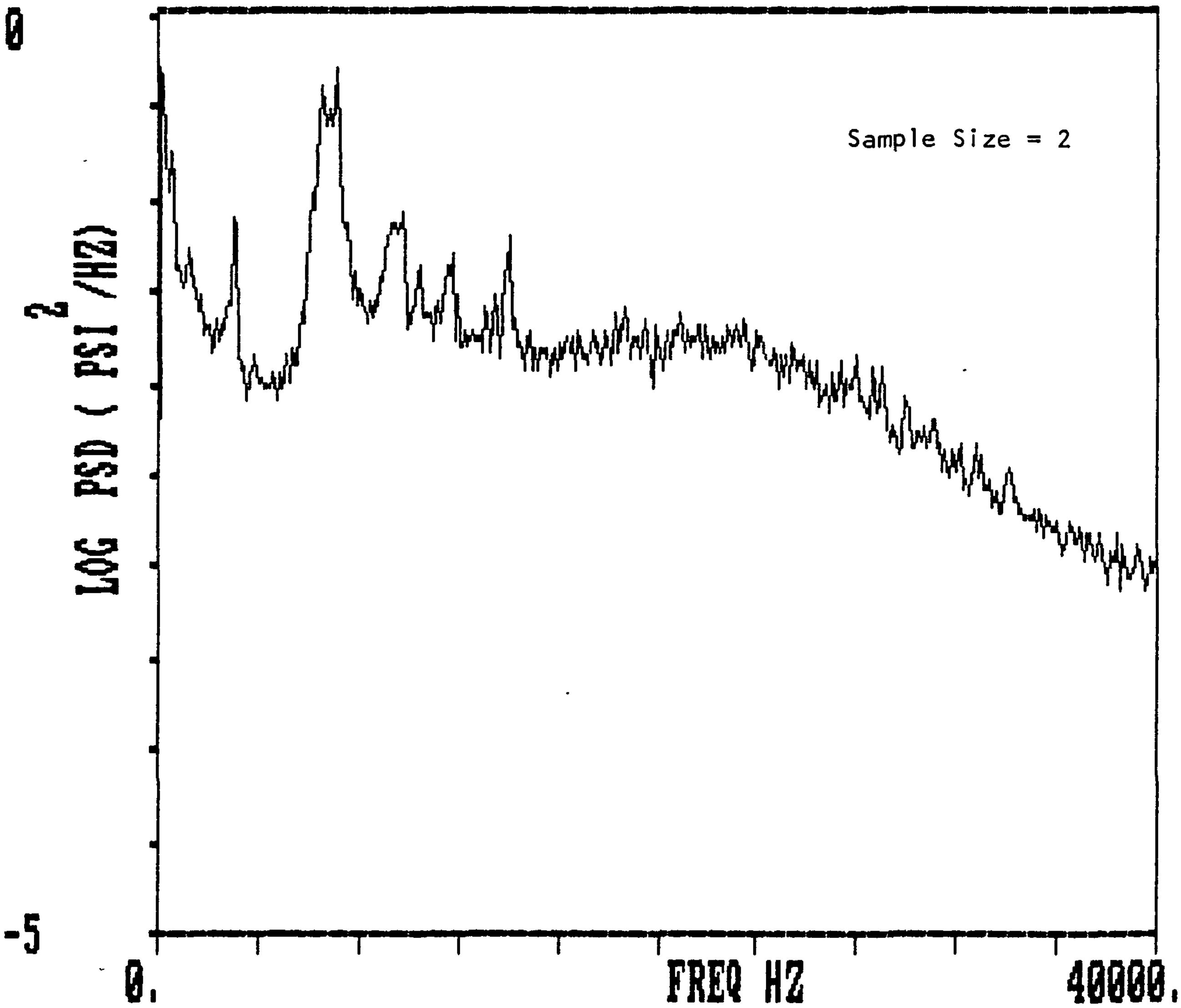


FIGURE 4-25. SPECTRUM OF MCC IN PR AT RPL

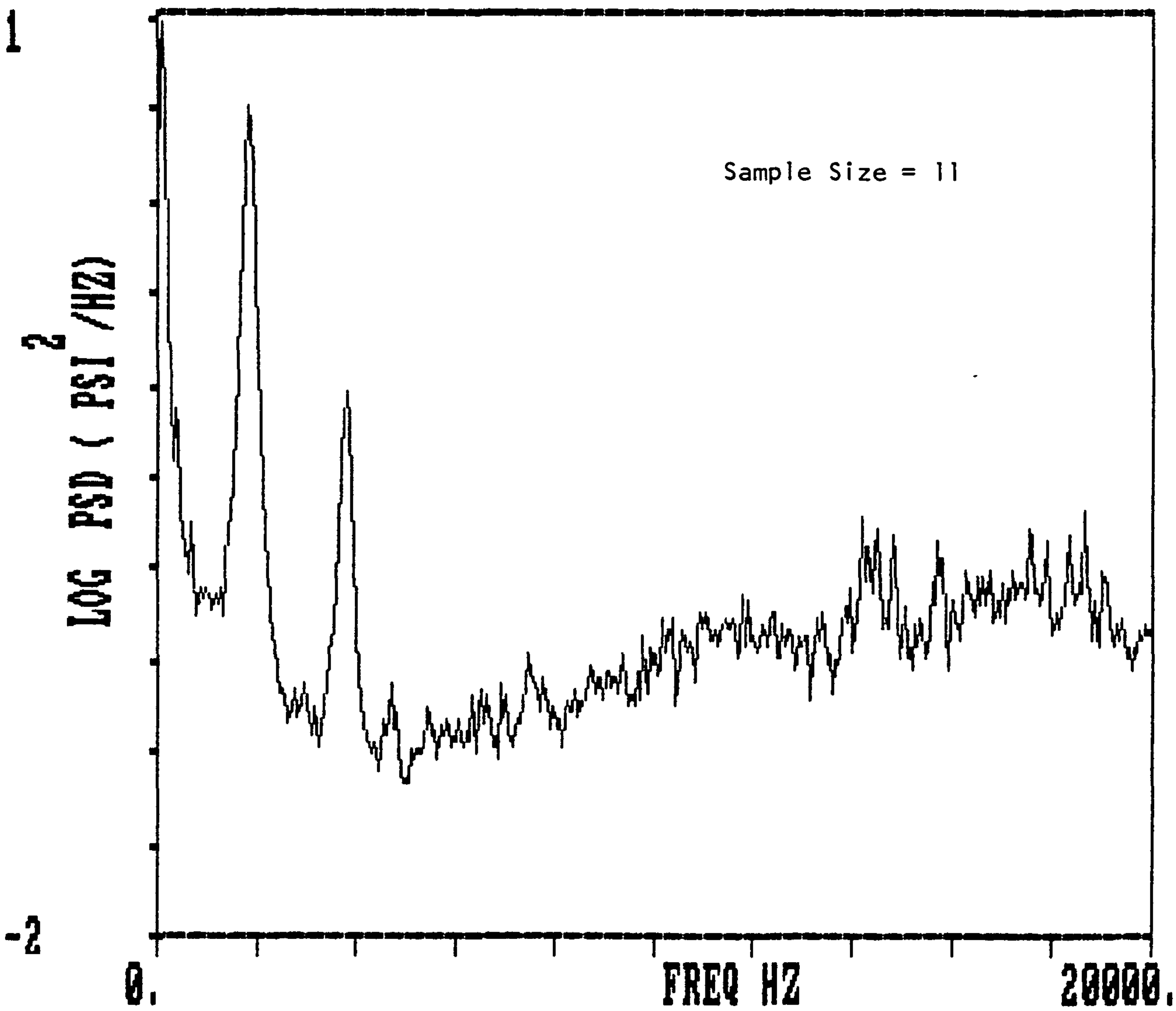


FIGURE 4-26. SPECTRUM OF OPB PC PR DC AT RPL

SECTION V

SPECTRUM NORMALIZATION AND INTERPOLATION

In the introduction to this report it was noted that the study was directed to accomplishment of three major objectives; data base generation, management software development, and development of techniques to estimate measurement spectral trends with engine operating conditions. This section summarizes approaches taken to accomplish the last objective.

5.1 Dimensional Analysis

At the outset of the study, it was anticipated that dynamic pressure measurement scaling would be accomplished through dimensional analysis based on governing fluid-dynamic and geometric parameters. A literature search was therefore initiated to define non-dimensional spectral relations for fluid/structural systems similar to SSME components. Two such non-dimensional spectral models are as follows:

Flow Induced Turbulence(1) (Duct Flow)

$$S_p = Q_p\ (s)/P^2 v^3 d$$

where S_p is the dimensionless pressure spectral density, and

Q_p = measured PSD

P = local fluid density

v = mean flow velocity

d = characteristic length

s= dimensionless frequency = fd/v (Strouhal number)

f = frequency

(1) Clinch, J.M. Prediction and Measurement of Vibrations Induced in Thin Walled Pipes by Internal Turbulent Waterflow. J. Sound Vib., V. 12, No. 4, 1970.

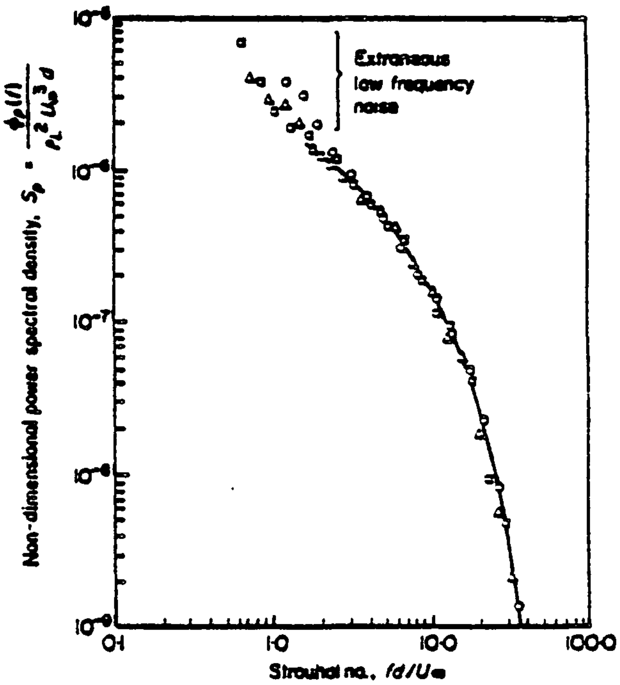


FIGURE 5-1. Frequency spectrum of turbulent wall pressure field. (○), 267 in/sec; (△) 450 in/sec; (□) 520 in/sec

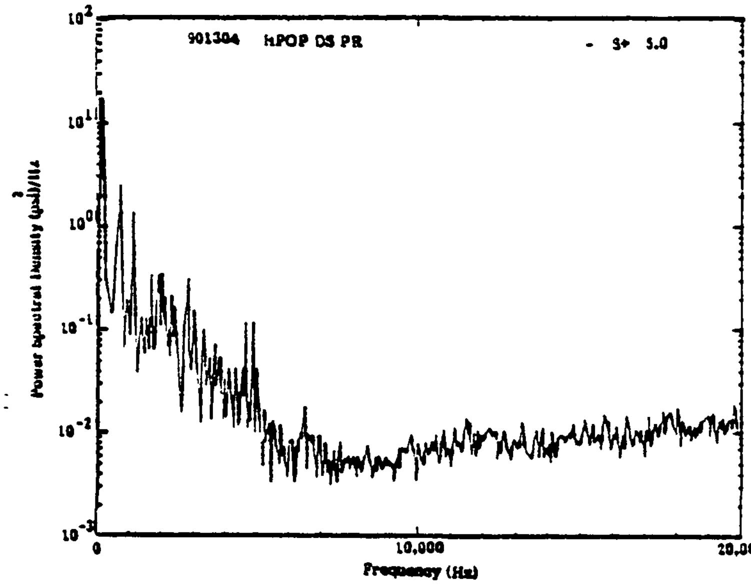


FIGURE 5-2. Power spectral density of the high pressure oxidizer pump discharge pressure 5 seconds after SSME startup

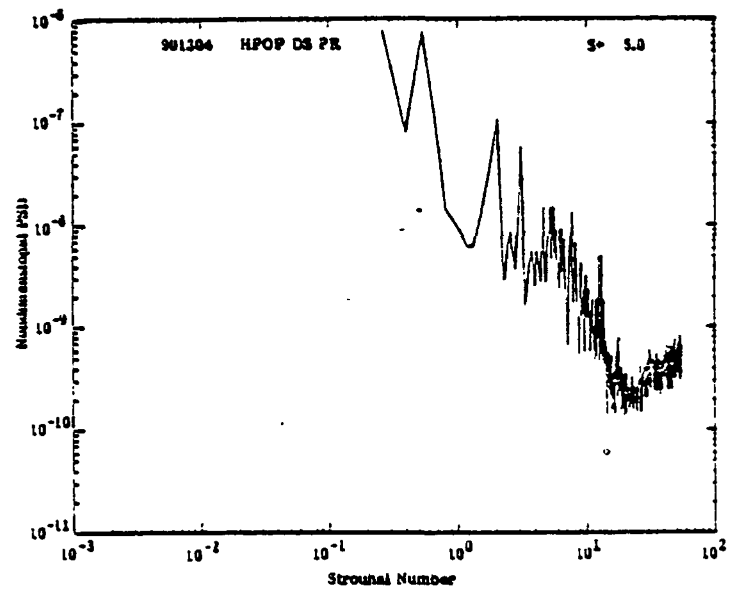


FIGURE 5-3. Nondimensionalized frequency spectrum for high pressure oxidizer pump discharge pressure, 5 seconds after SSME startup

Blade Wake Excitation(2) (Pump Discharge)

$$g(s, \ ) = Q_p/Pc^2D^3M_t{}^3$$

where

	= Flow coefficient
M_t	= Blade tip mach number
s	= Dimensionless frequency, f/n
D	= Blade Diameter
n	= Shaft speed

Figure 5-1, from reference (1), illustrates the non-dimensional wall pressure spectrum measured on a circular duct with internal water flow over a range of flow velocities. Collapse of the data is seen to be quite good. Figure 5-2 is the ordinary PSD of a dynamic pressure measurement. Dimensional normalization of this spectrum by the above technique is shown in figure 5-3. The predominant spectral peaks appear related to turbopump rotational speeds. A normalization scheme was therefore developed to represent the spectra as a linear superposition of simple non-dimensional sources.

As sufficient data base measurements became available a detailed evaluation of spectral trends with engine power level was performed. (It should be noted that the non-dimensionalizing parameters such as flow velocity and turbopump blade tip speed vary in proportion to engine power level). Since well defined spectral trends were not indicated in general, it became clear that the measured flow dynamic interactions are too complex to be adequately represented by a superposition of simple sources.

(2) Maling, G. C. Jr. Dimensional Analysis of Blower Niose. J. Acoust. Soc. Am., V. 35 No. 10, 1963.

Dimensional analysis was therefore abandoned in favor of a more general two dimensional interpolation procedure.

5.2 Interpolation By Singular Value Decomposition

The method of Singular Value Decomposition is a mathematical tool for the decomposition of general rectangular matrices. By this technique, a family of curves, such as the power spectra of dynamic pressure measured at a number of power levels can be reduced to a superposition of basic component patterns (eigenvectors) and associated weighting coefficients. The predominant normalized component patterns yield a good approximation of the given spectra. Thus, the difficult problem of interpolating (two dimensional) over the observed spectra can be reduced to one dimensional polynominal interpolation of the derived weighting coefficients.

It may be instructive to observe that the SVD technique is mathematically akin to the normal mode method in structural dynamics. In the latter application, the complex deflection patterns of a structure are expressed as a superposition of normal modes, or eigenvectors. The associated weighting coefficients (eigenvalues) are the system natural frequencies. (An excellent review of the SVD technique, upon which this discussion and the resulting software is based, is given by S. Tominaga in the IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-29, No. 3, June, 1981.) The technique is best illustrated by example.

Figure 5-4 illustrates the ensemble average spectra of the high pressure oxidizer pump discharge pressure observed at 11 engine power levels. Based on this set of data we wish to extract normalized spectrum functions and associated coefficients to represent the curves over a continuous range of power levels. Each PSD may be represented as a vector p,(n x 1), representing the spectral values at n

frequencies. The group of spectra may thus be written as a matrix P, (n x m), representing the m spectra at n frequencies. (In the case at hand this is a 400 x 11 matrix.) If the rank of P is m, it may be expressed as the sum of m matrices of rank one,

$$P = r_1 n_1 e^T{}_1 + \ldots + r_m n_m e^T{}_m$$

where n_i, e_i are the orthonormal vectors of PP^T and P^TP, respectively, and the singular values r_i are the square roots of the eigenvalues of PP^T.

Now, each of the eleven spectra may be uniquely defined as a linear combination of the fundamental component waves n_i, in the form

$$p_i = c_{i1}n_1 + \cdots + c_{im}n_m$$

where $c_{ij} = r_j e_{ij}$ and e_j is defined to be $[e_{1j}, \ldots e_{mj}]^T$.

The SVD technique provides a method of synthesizing a set of observed curves. For purposes of prediction it is then only necessary to interpolate over the coefficients c_{ij}, as a function of engine power level, for each fundamental component pattern.

The software was developed for the implementation of the SVD technique and interpolation scheme. A program listing of the routines is given in Table 5-1. Figure 5-5 illustrates the fundamental component patterns associated with the three largest singular values for the spectra in Figure 5-4.

The above analysis indicates that all component patterns be utilized to synthesize the family of curves. This provides a unique representation. In practice, however, convergence is quite rapid. This is again similar to the normal mode

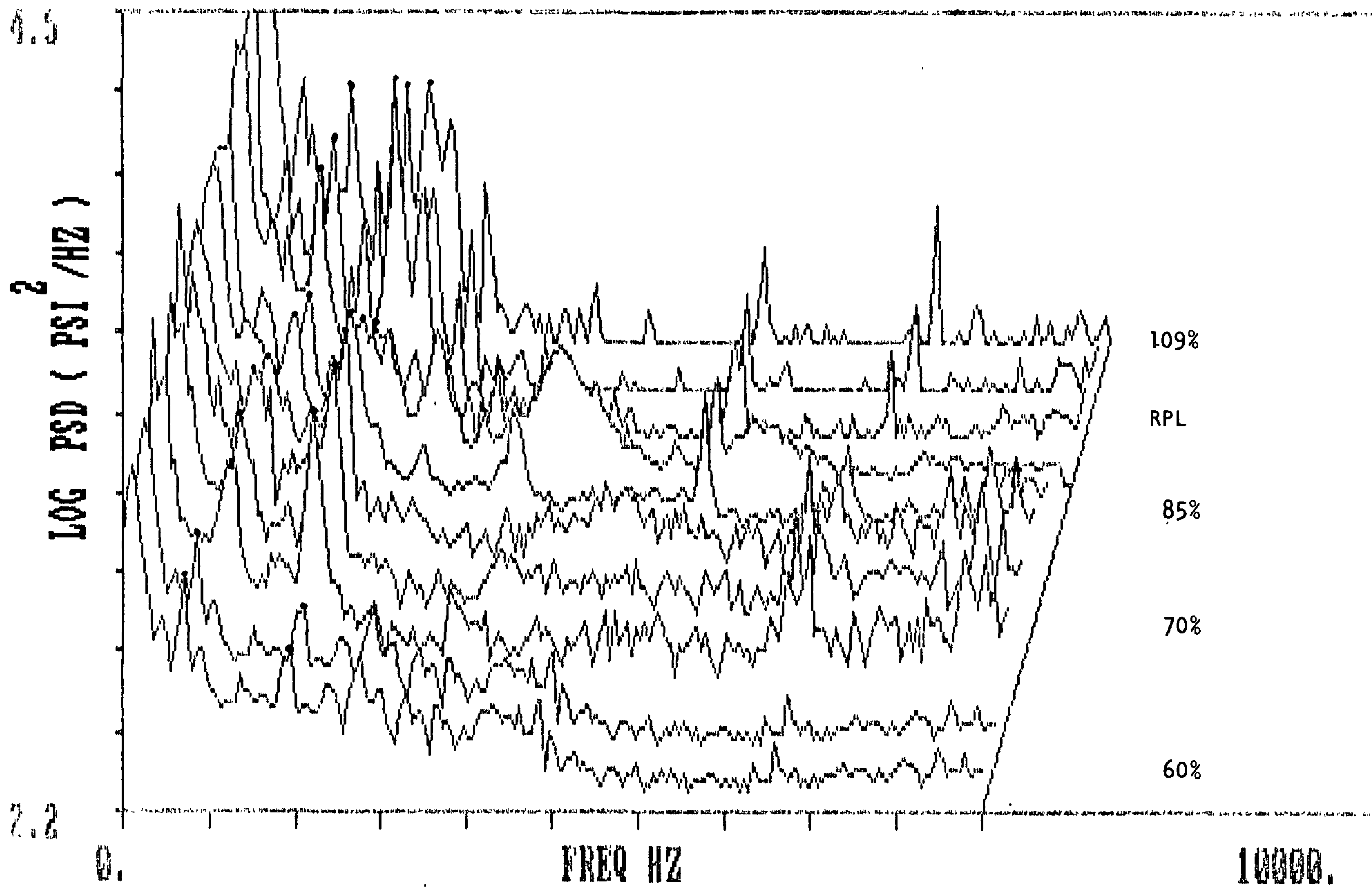


FIGURE 5-4. ENSEMBLE AVERAGE SPECTRA AS A FUNCTION OF ENGINE POWER LEVEL (HPOP DS PR)

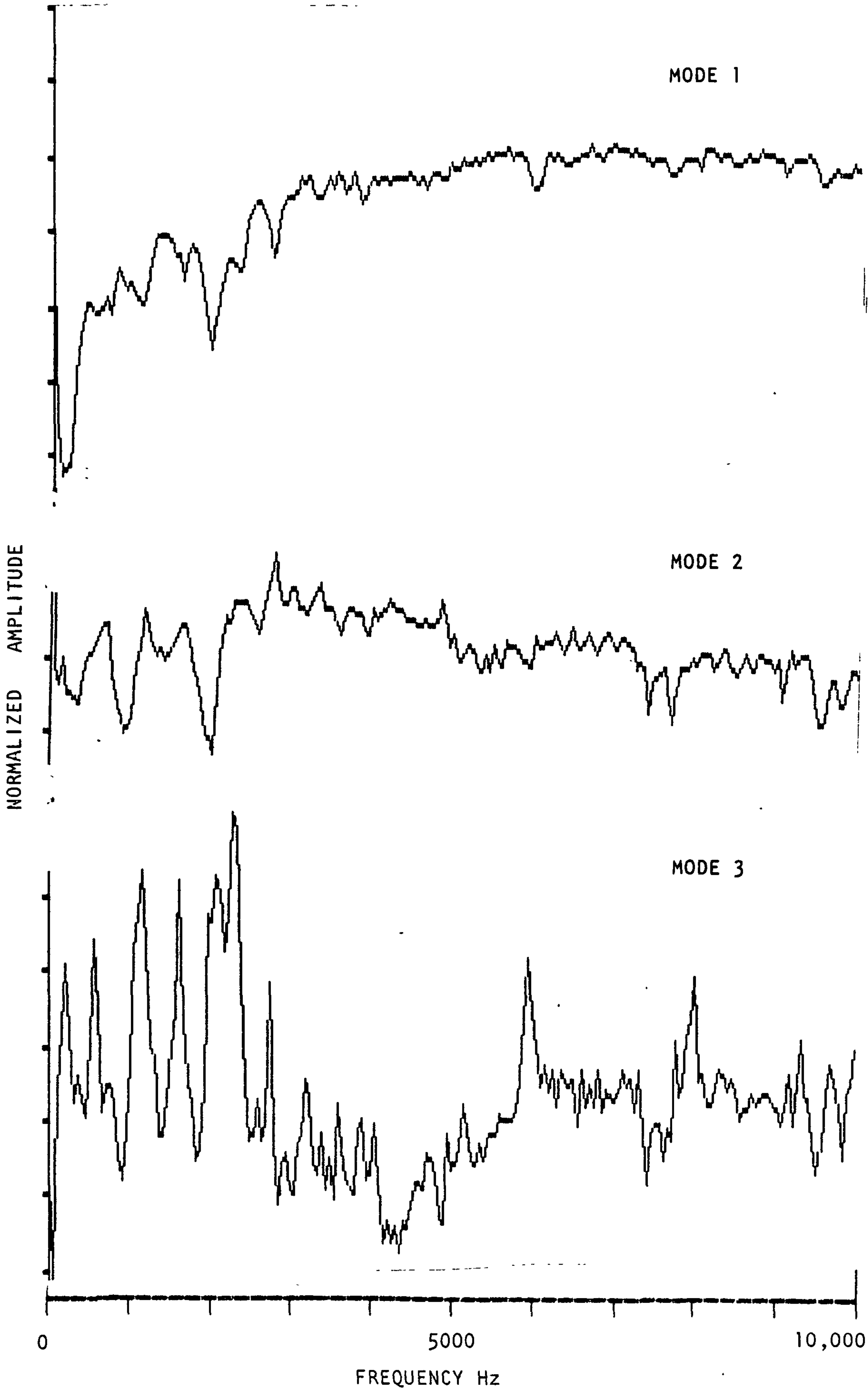


FIGURE 5-5. NORMALIZED SPECTRUM FUNCTIONS (EIGENVECTORS)

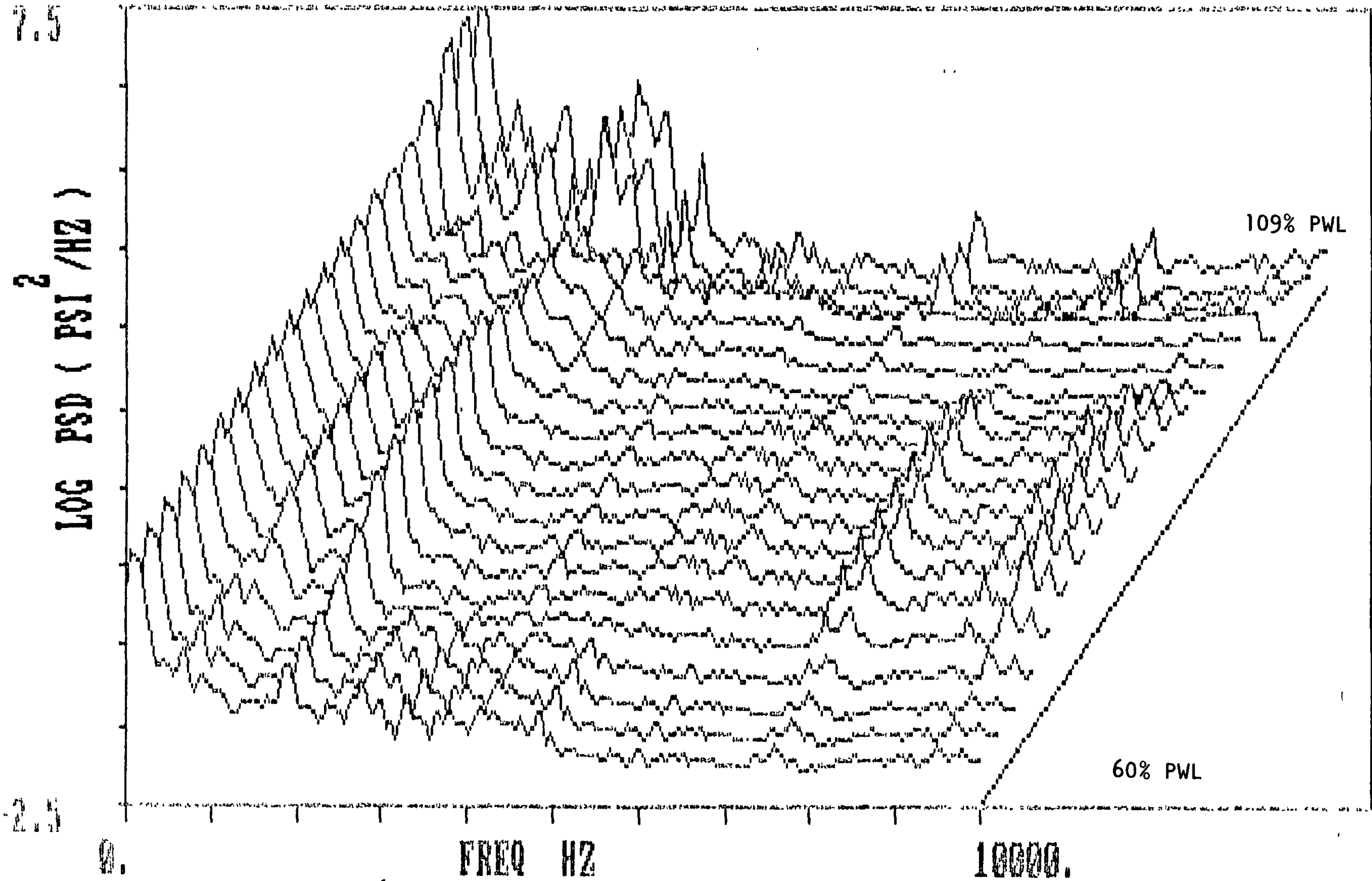


FIGURE 5-6. INTERPOLATED SPECTRAL TREND BY SINGULAR VALUE DECOMPOSITION (2.5% PWL INCREMENTS)

method, wherein the deflection of a beam can generally be adequately represented by the superposition of a limited number of modes.

For the example under discussion, the eleven computed singular values (we have eleven PSD's) in descending order are: 105.3, 9.8, 6.5, 4.4, 3.9, 3.4, 3.3, 2.9, 2.7, 2.3 and .00002. The relative contribution of each fundamental pattern may be shown to be proportional to its associated singular value. It is therefore clear that a superposition of the first three fundamental patterns should provide a good approximation to the given family of spectra. This was done for the cited example, and an interpolated library of pressure spectra was computed at 2.5% power levels over the range of observed test values. These results are shown in Figure 5-6, and indicate the spectral trend of dynamic pressure with engine power level.

5.3 An Interpolated Library of Spectra for Dynamic Pressure Measurements

To indicate the trend in amplitude/frequency of the SSME dynamic pressure measurements an interpolated library of spectra was derived, by the method just discussed. The analysis procedure was performed as follows:

1. The data base spectra for each measurement location were plotted, and any obvious outliers discarded from subsequent analysis.

2. The remaining data sets were segregated by power level. The range and number of power levels available varied by measurement location, based on the specific measurements and test conditions provided by MSFC.

3. The spectra for each measurement/power level were ensemble averaged to obtain a family of spectra as shown in the example, Figure 5-4.

4. An SVD analysis was performed on each set of curves to extract intrinsic component patterns and associated singular values. The number of derived patterns/coefficients varied according to the number of test power levels available for each measurement.

5. Finally, interpolation was applied to the above derived coefficients, and plots generated representing spectral variation in 2.5% power level increments over the available data range.

The results of this exercise are illustrated in the remaining figures of this report. These figures provide a qualitative indication of spectrum variation with engine power level. A quantitative estimate of spectral amplitudes and frequencies may be obtained by cross reference to the average spectra representing operation at RPL, as given in Section IV of this report.

It will be noted that several of the data base measurements are not included in the above discussed spectrum library. This is essentially due to lack of sufficient data; either a very limited number of spectra and/or a very limited test power level range. In either case, one is referred to the average spectra presented in Section IV or the data base plots provided in the Appendices for spectral definition of these measurements.

5.4 A Computer Routine for Estimating Dynamic Pressure Environments

The interpolated spectrum libraries presented in the previous discussion provide a valuable indication of spectral trends with engine power level. However, as noted, the results are qualitative. A computer routine was therefore developed to generate an estimated spectrum for a user defined dynamic pressure measurement and engine power level.

This program utilizes the eigenvectors and singular values obtained by the SVD technique as discussed in Section 5-3 for each measurement, and is therefore totally indepedent of the complete data base. The routine was designed for the engineer responsible for dynamic evaluations and requires no knowledge of the linear algebra implied by SVD.

A simple menu driven format is provided, wherein the user is querried as to what measurement description and power level a PSD estimate is desired for. (The user is also asked how many eigenvectors (fundamental SVD patterns) he/she desires to use in the spectrum estimate. To answer this question the user has the option to review the singular value magnitudes associated with each pattern. However, through development of the SVD analysis and the pressure spectrum libraries, experience indicates that a "three mode" approximation should be satisfactory and efficient.) A program listing of the spectrum estimation routine is given in Table 5-1. The program is operational on any PC/compatible computer.

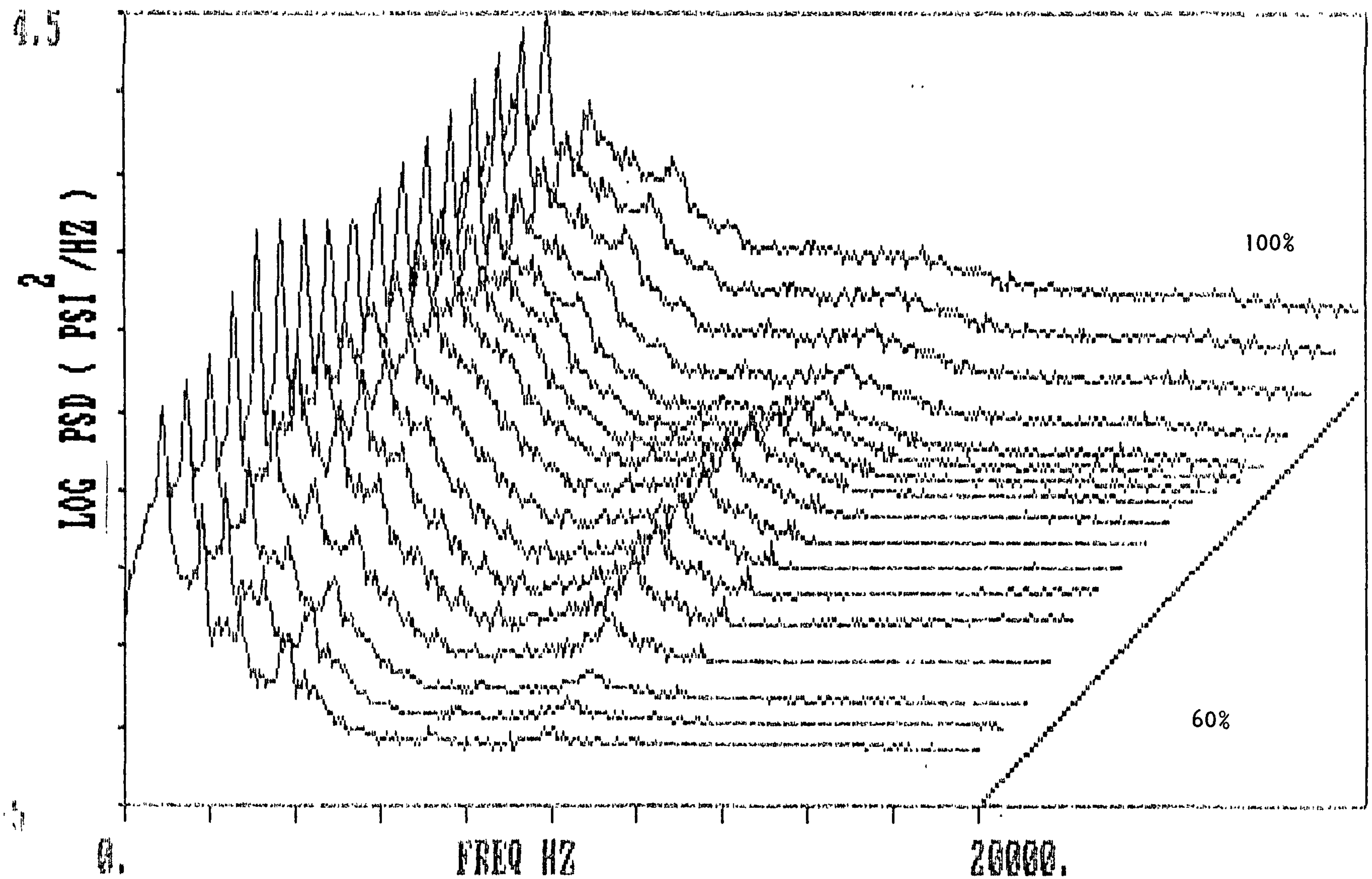


FIGURE 5-7. SPECTRAL TREND OF LPFP IN PR (60 - 100% RPL, 2.5% INCREMENTS)

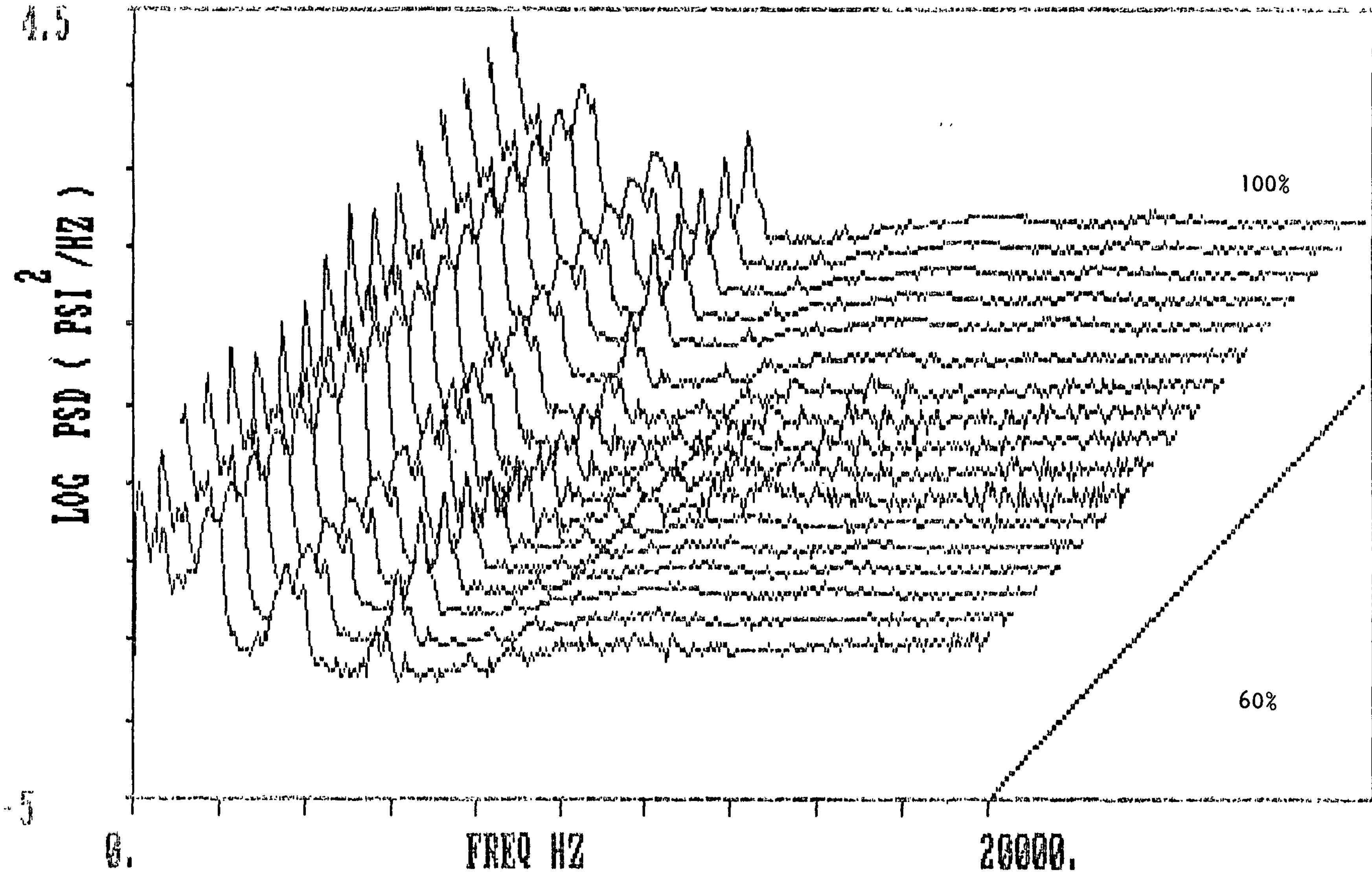


FIGURE 5-8. SPECTRAL TREND OF LPOP DS PR (60 - 100% RPL, 2.5% INCREMENTS)

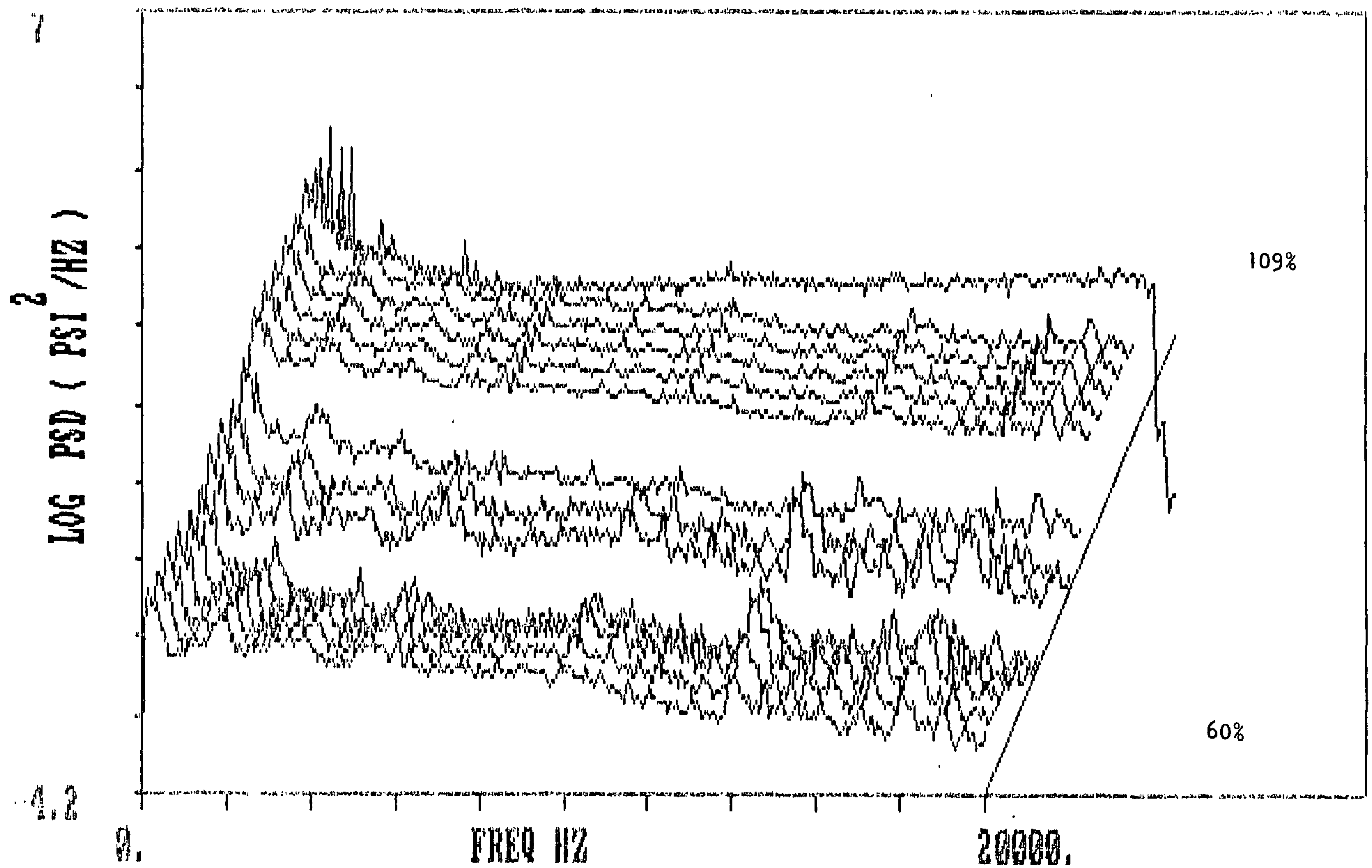


FIGURE 5-9. SPECTRAL TREND HPOP BAL CAV PR2 (60 - 109% RPL)

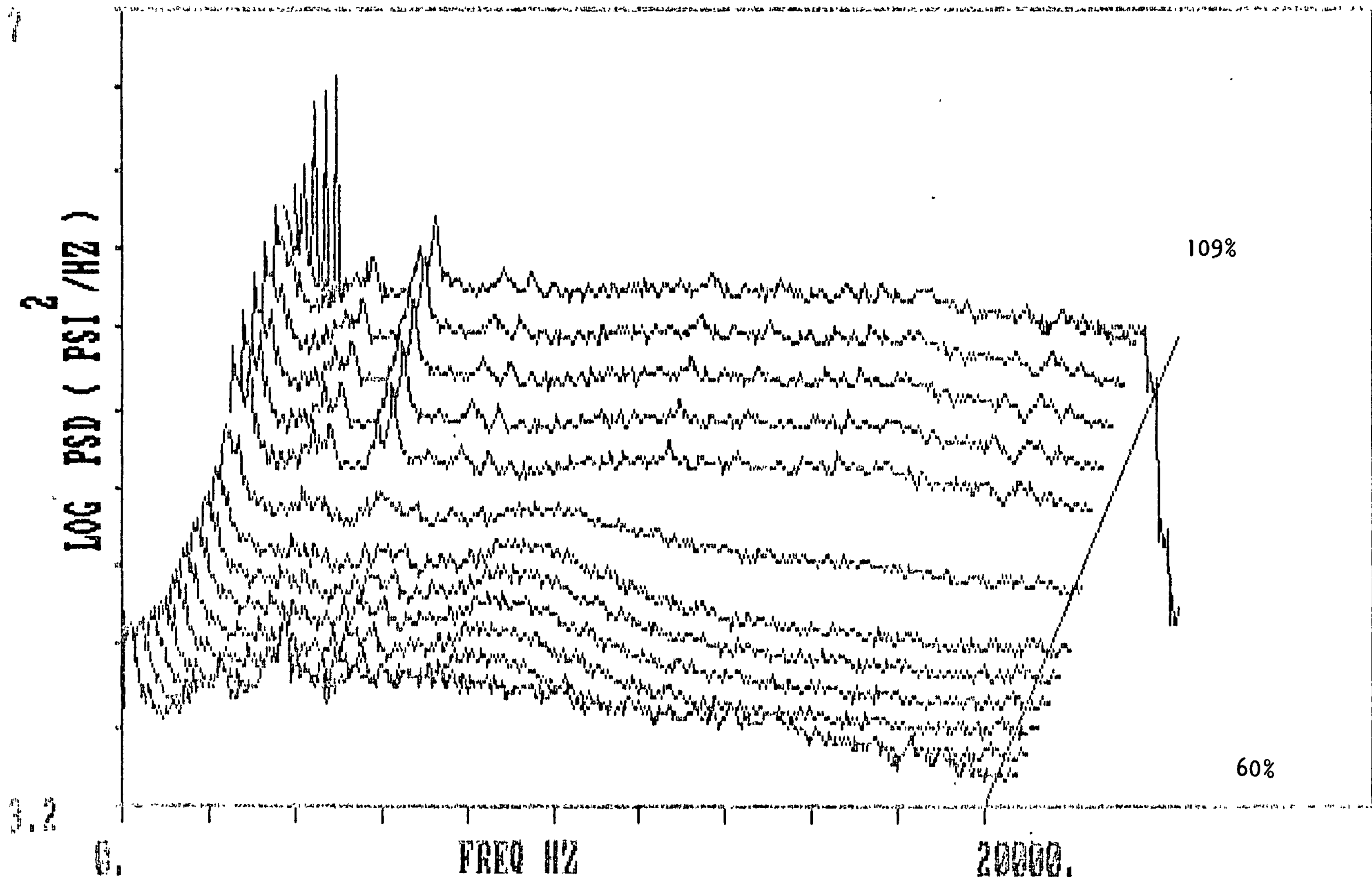


FIGURE 5-10. SPECTRAL TREND OF HPOP BAL CAV PR1 (60 - 109% RPL)

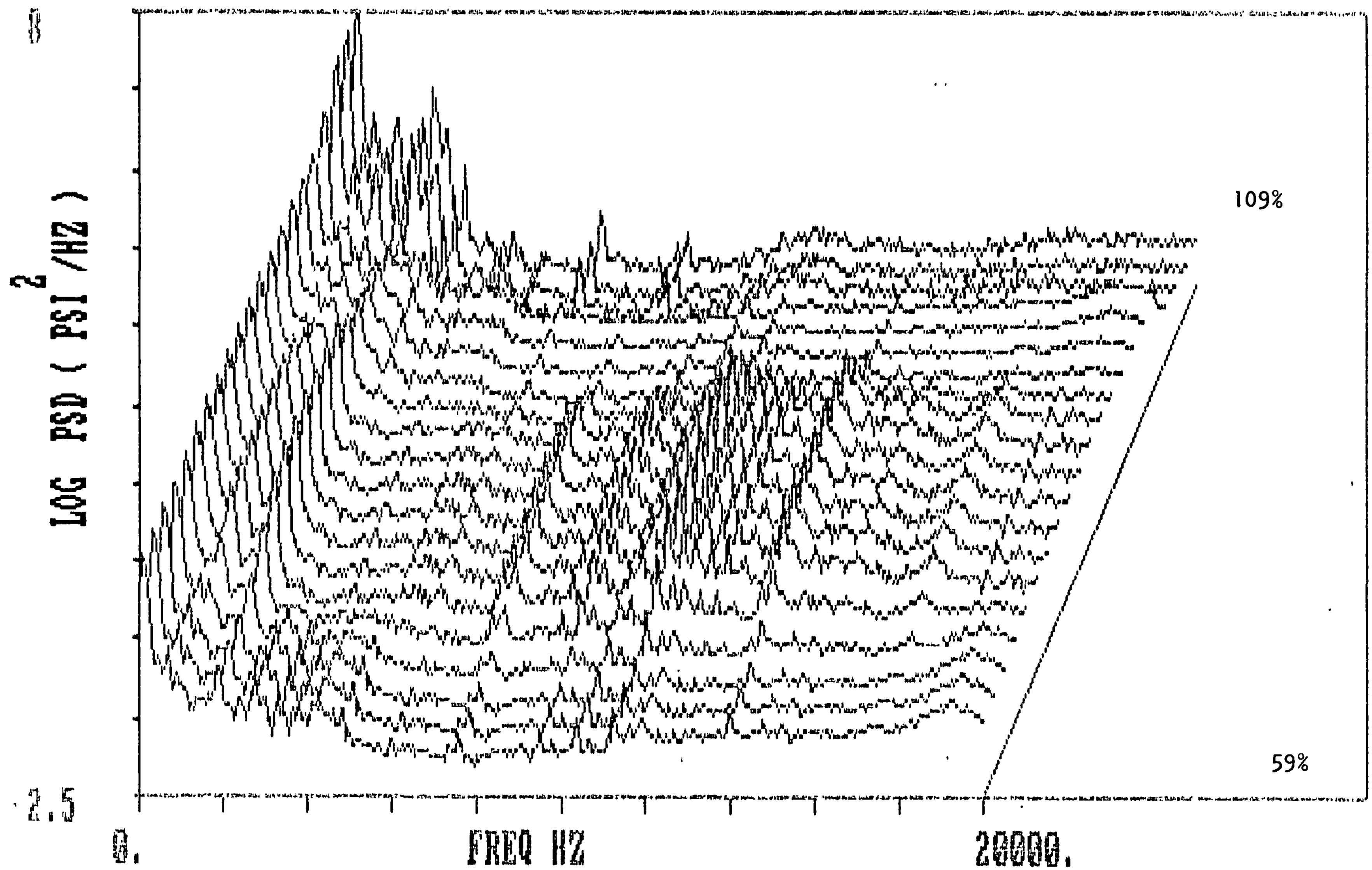


FIGURE 5-11. SPECTRAL TREND HPOP DS PR (59 - 109% RPL, 2.5% INCREMENTS)

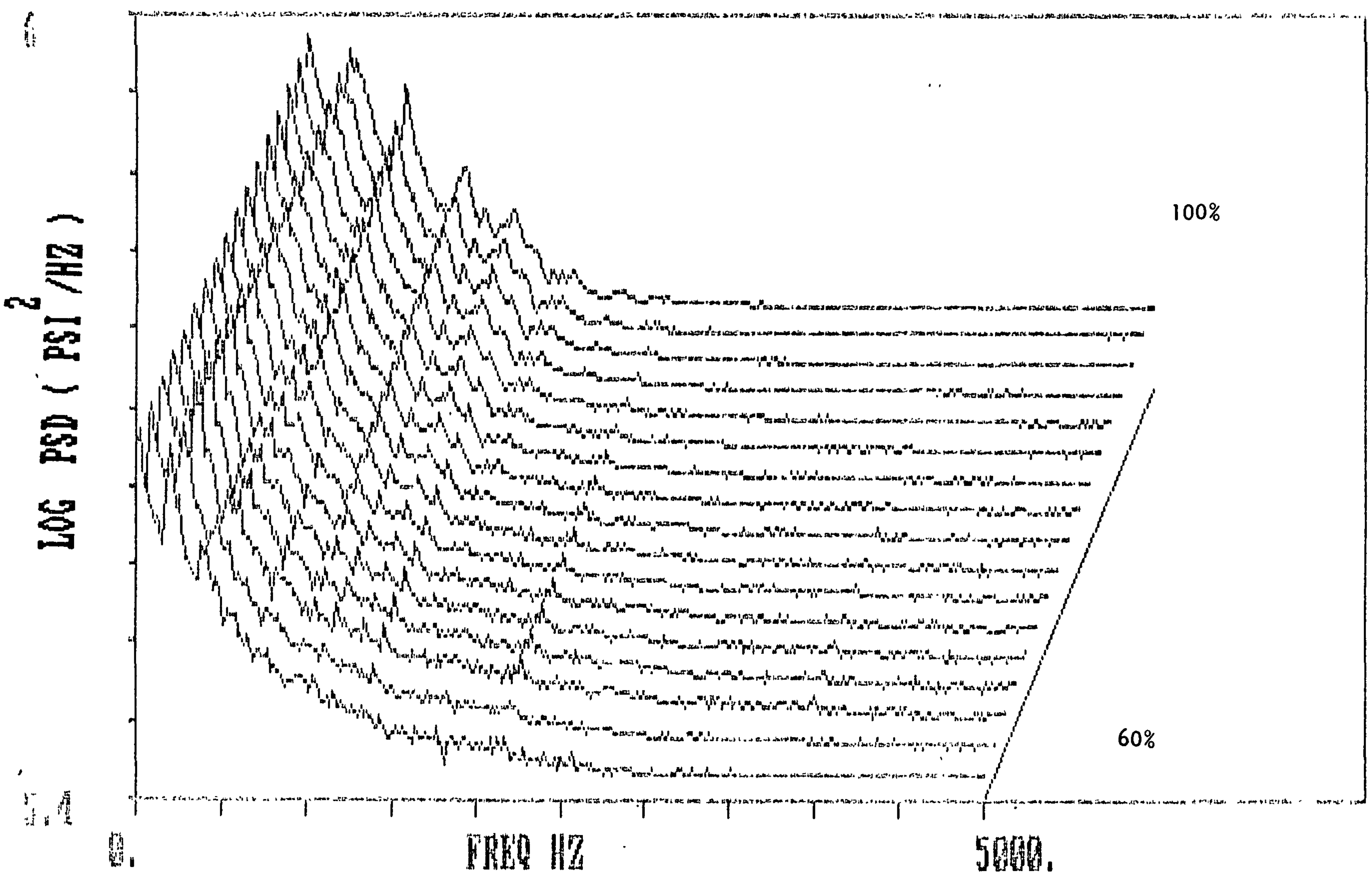


FIGURE 5-12. SPECTRAL TREND LOX IN DUCT PR2 (60 - 100% RPL)

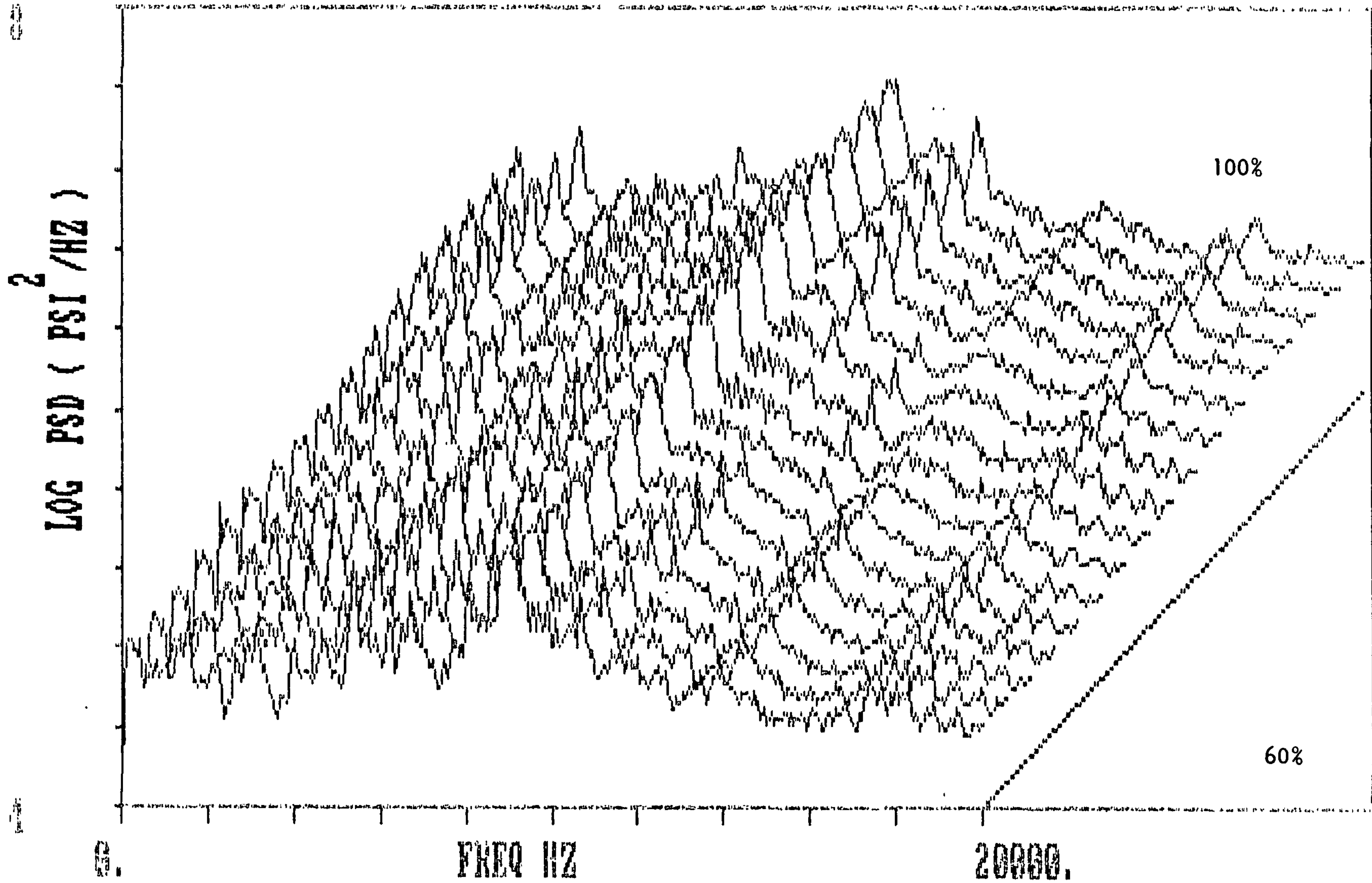


FIGURE 5-13. SPECTRAL TREND OF HPOP DISC PR (60 - 100% RPL)

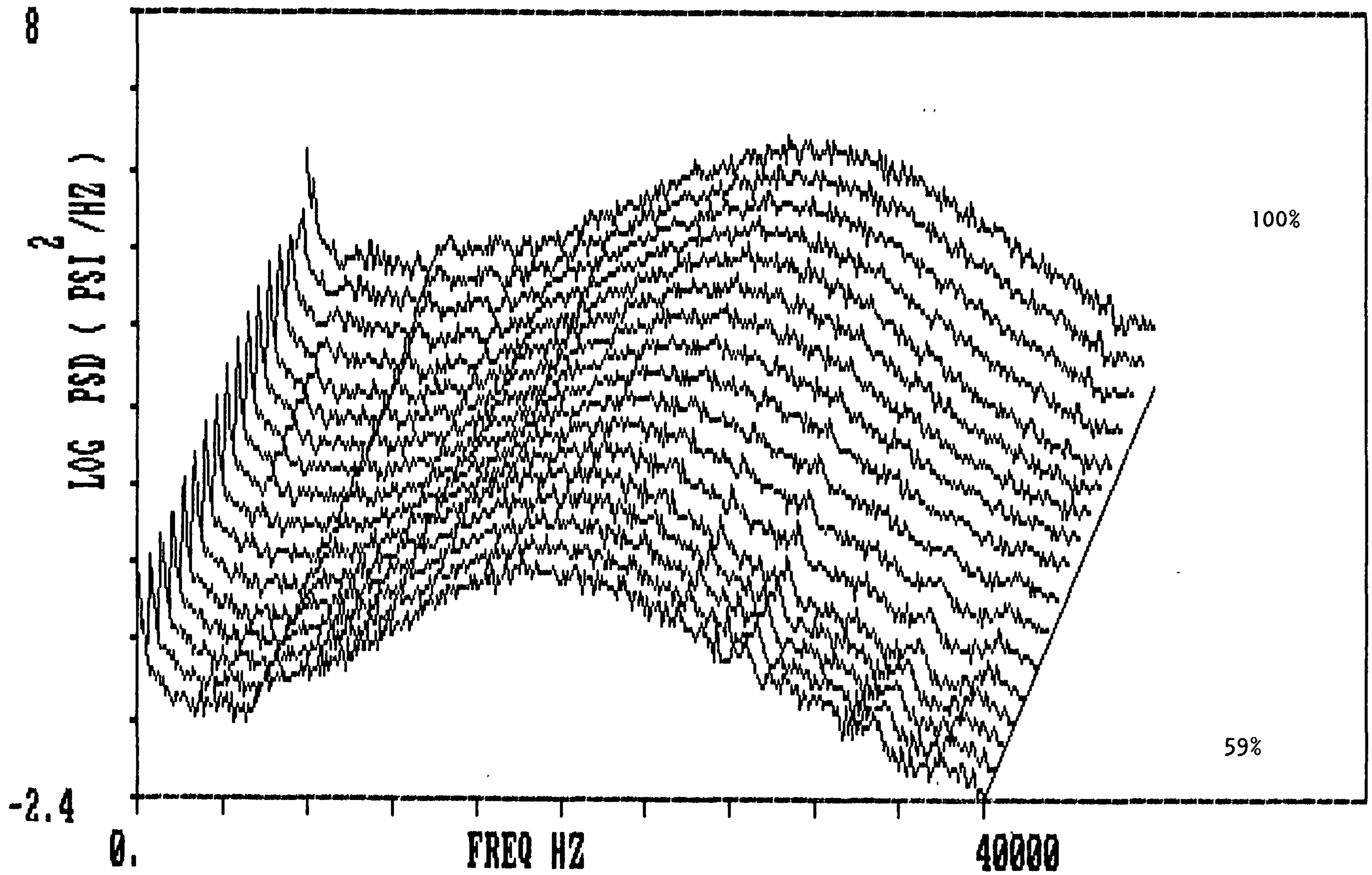


FIGURE 5-14. SPECTRAL TREND MCCHOT GAS IN PR (59 - 100% RPL)

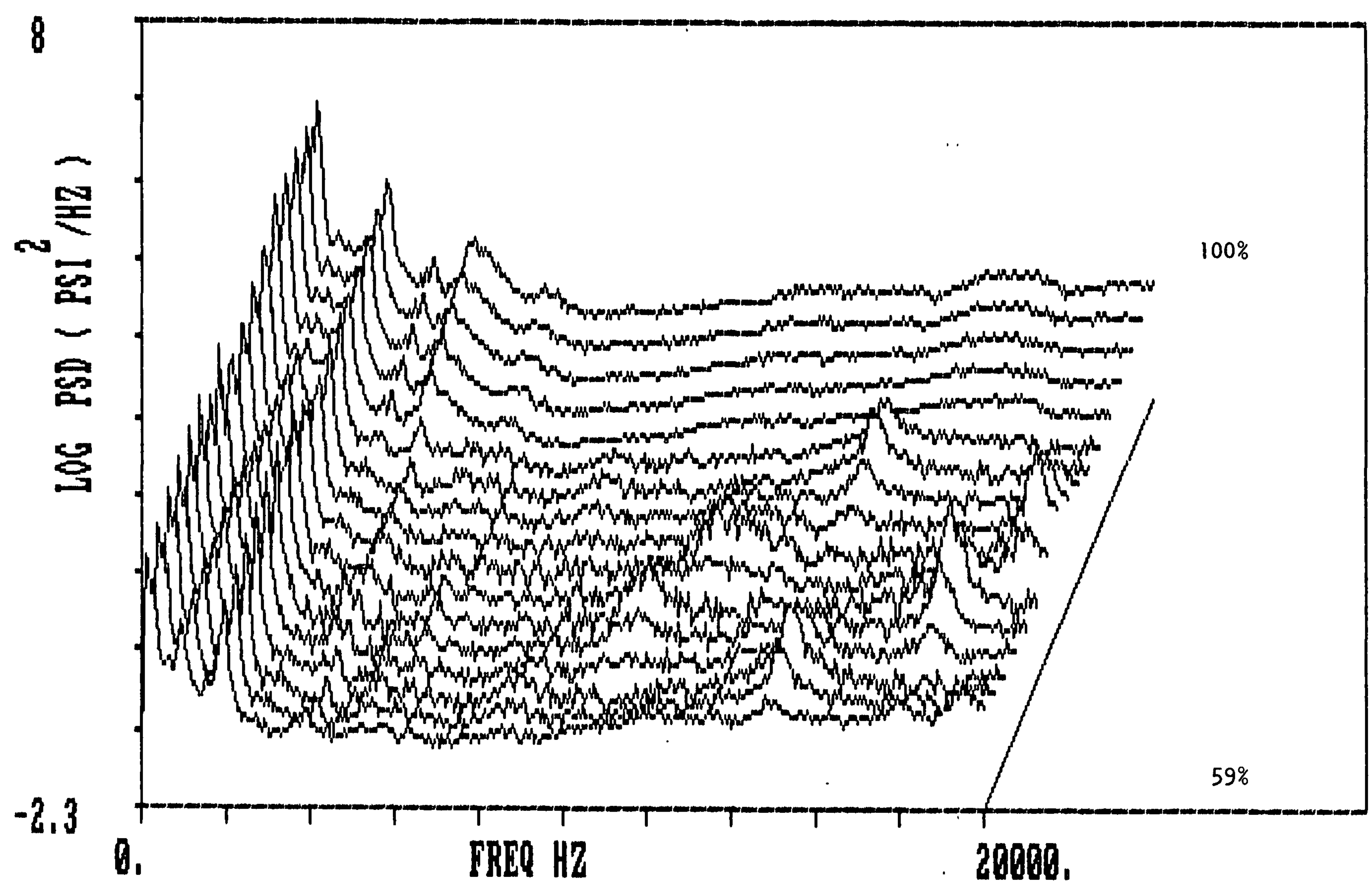


FIGURE 5-15. SPECTRAL TREND PBP DS PR (59 - 100% RPL)

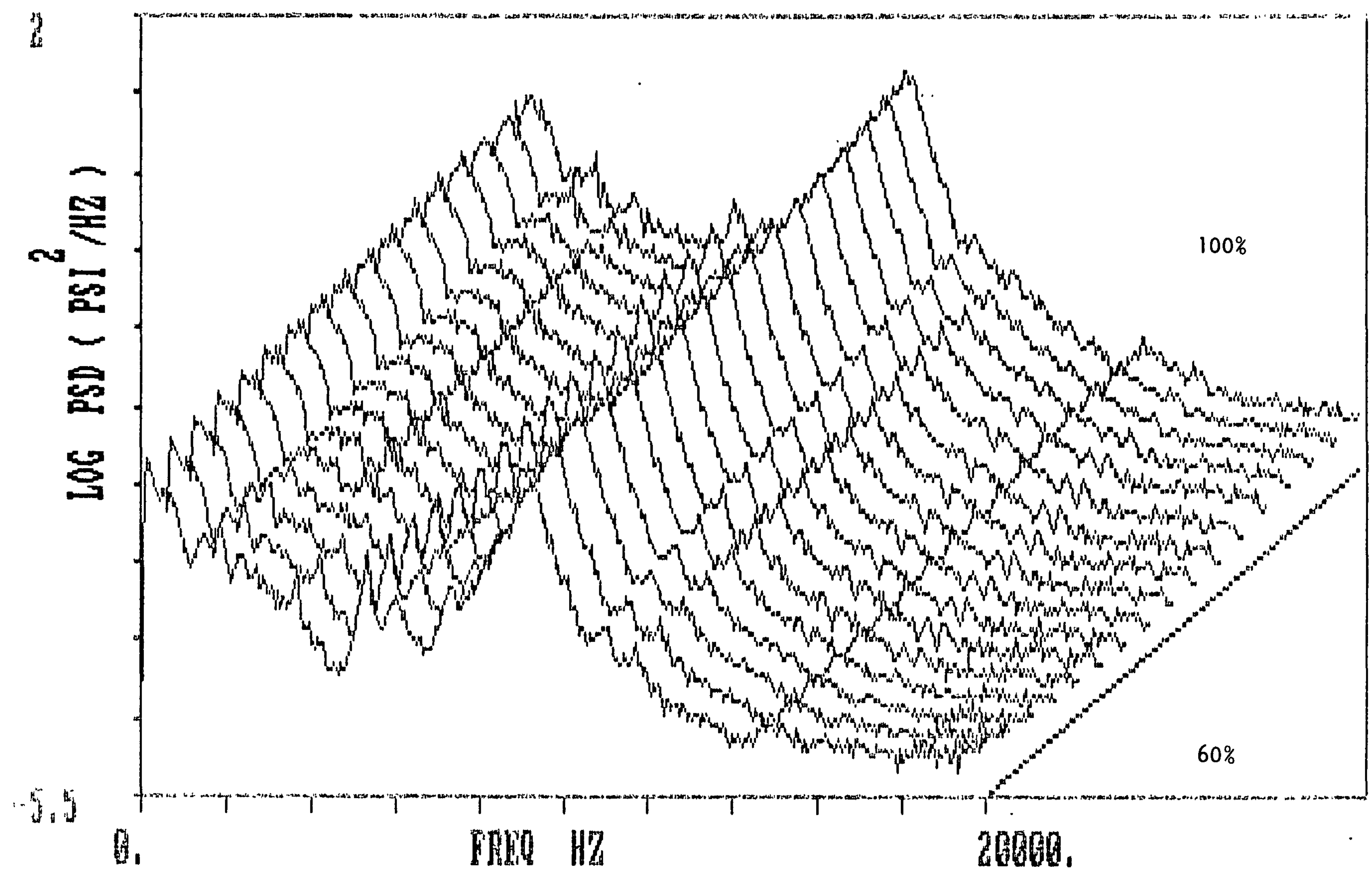


FIGURE 5-16. SPECTRAL TREND LPFP DS PR (60 - 100% RPL, 2.5% INCREMENTS)

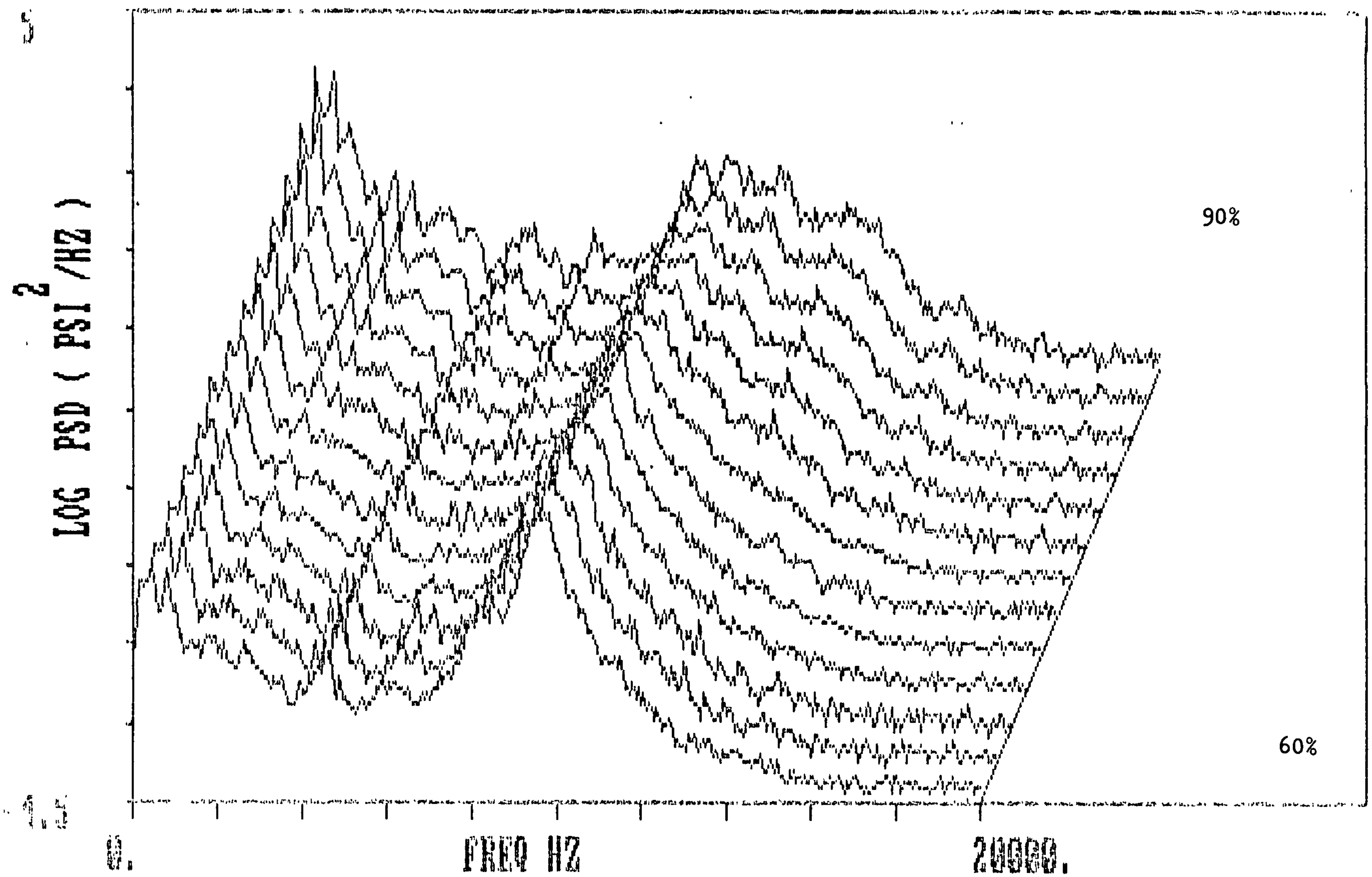


FIGURE 5-17. SPECTRAL TREND PBP FUEL SUB PR (60 - 90% RPL)

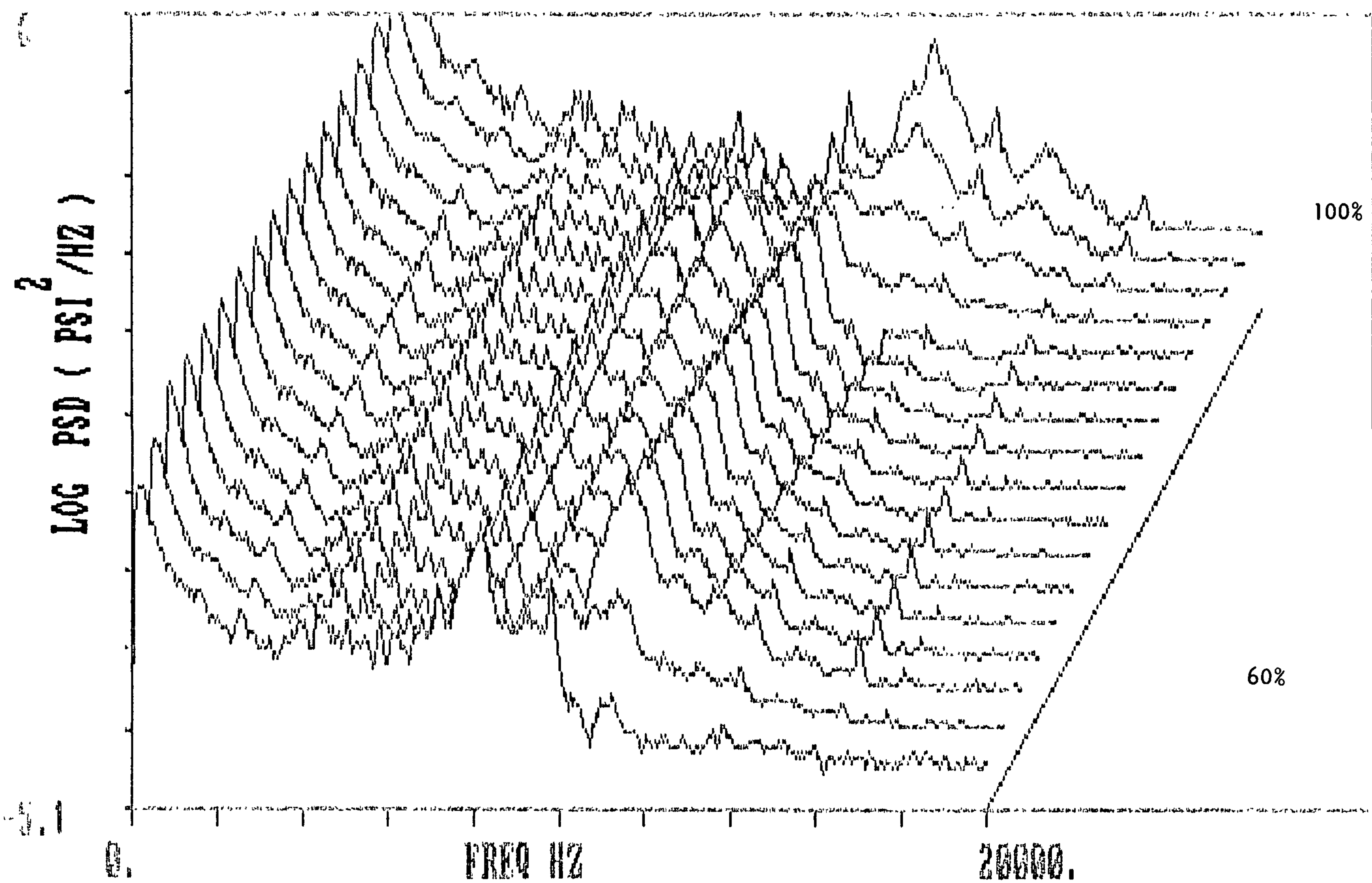


FIGURE 5-18. SPECTRAL TREND FPB FUEL MAN PR (60 - 100% RPL)

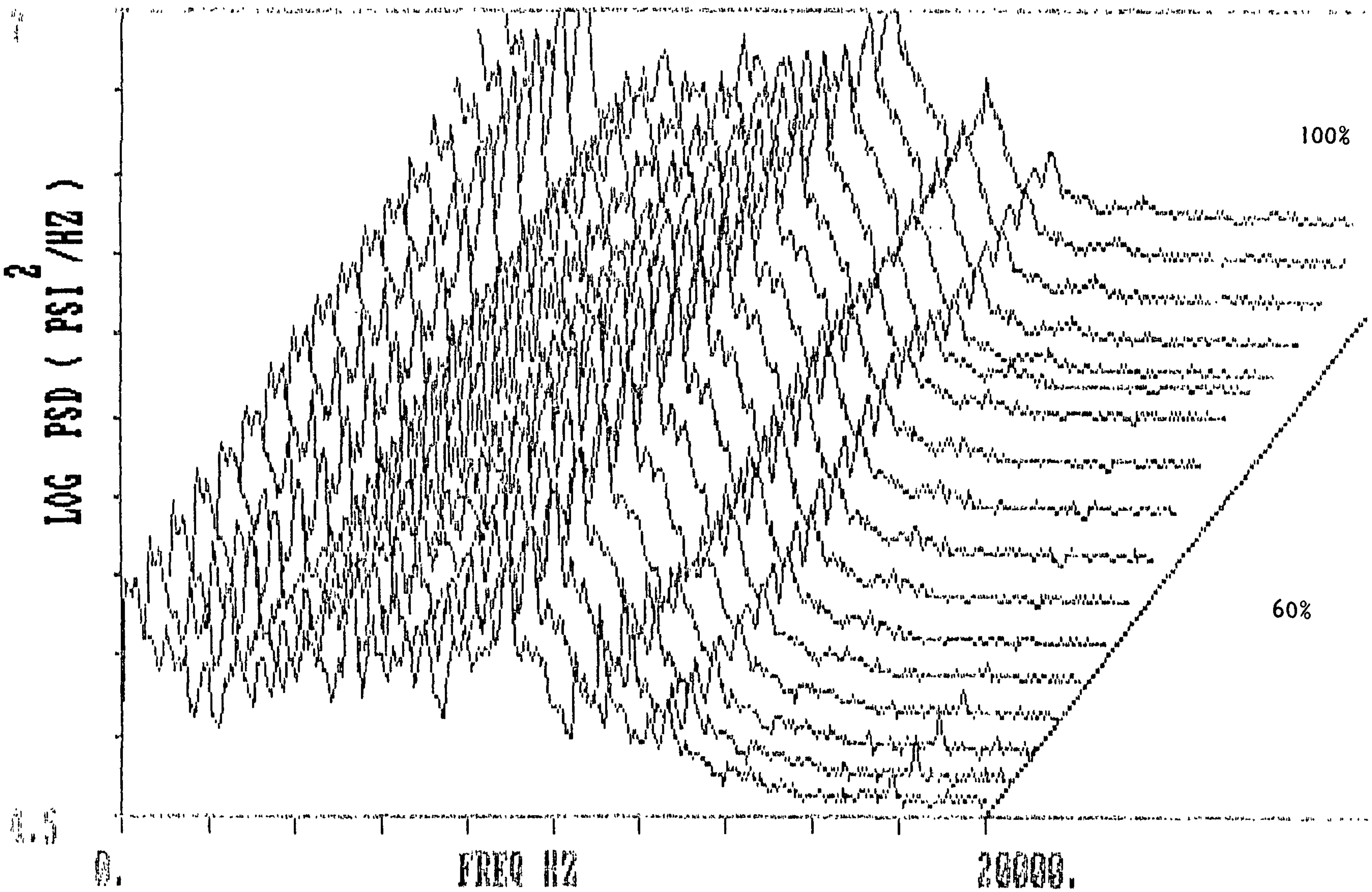


FIGURE 5-19. SPECTRAL TREND LPOT TURB DR PR (60 - 100% RPL)

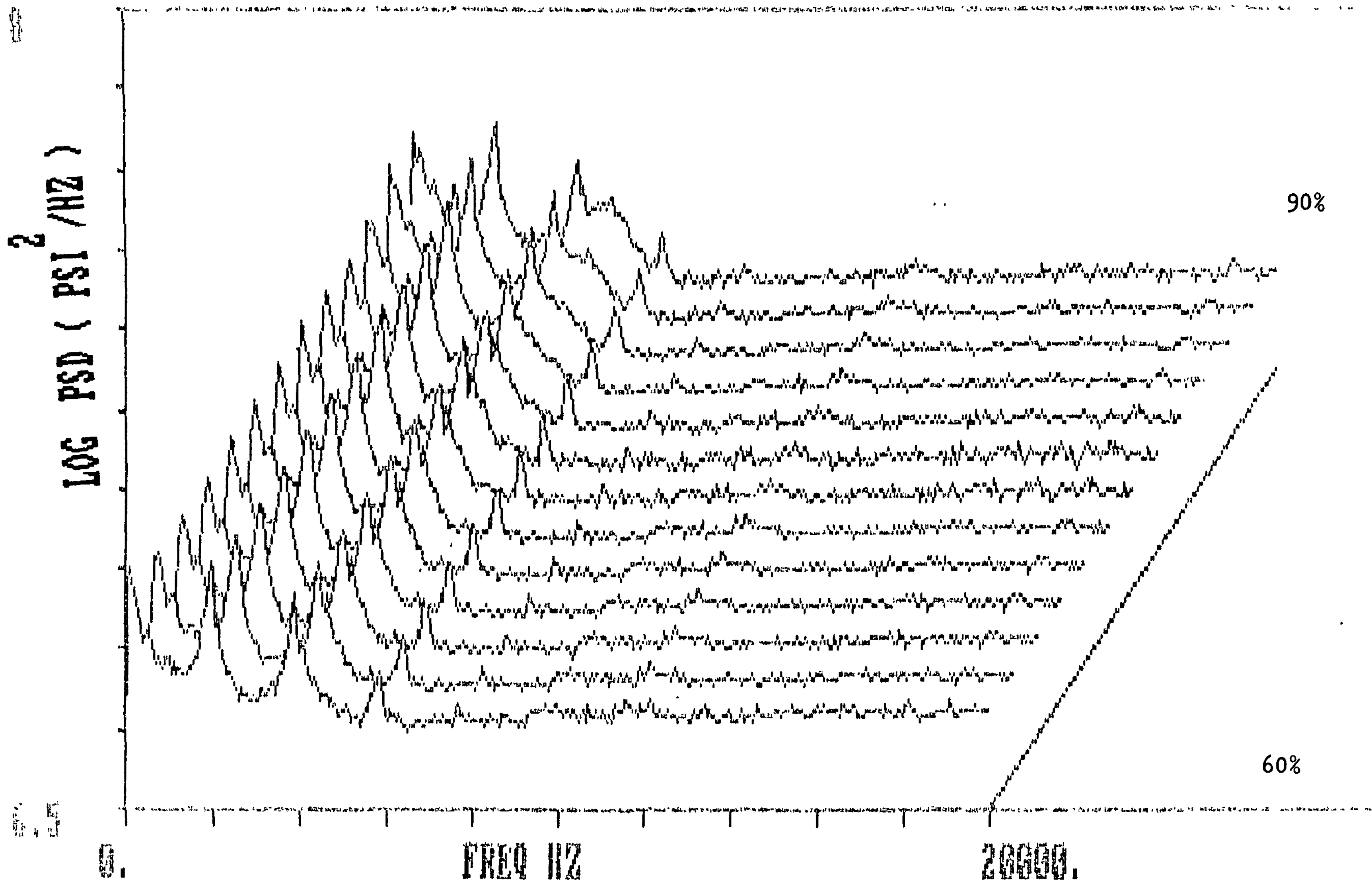


FIGURE 5-20. SPECTRAL TREND LOX HX OUT PR (60 - 90% RPL)

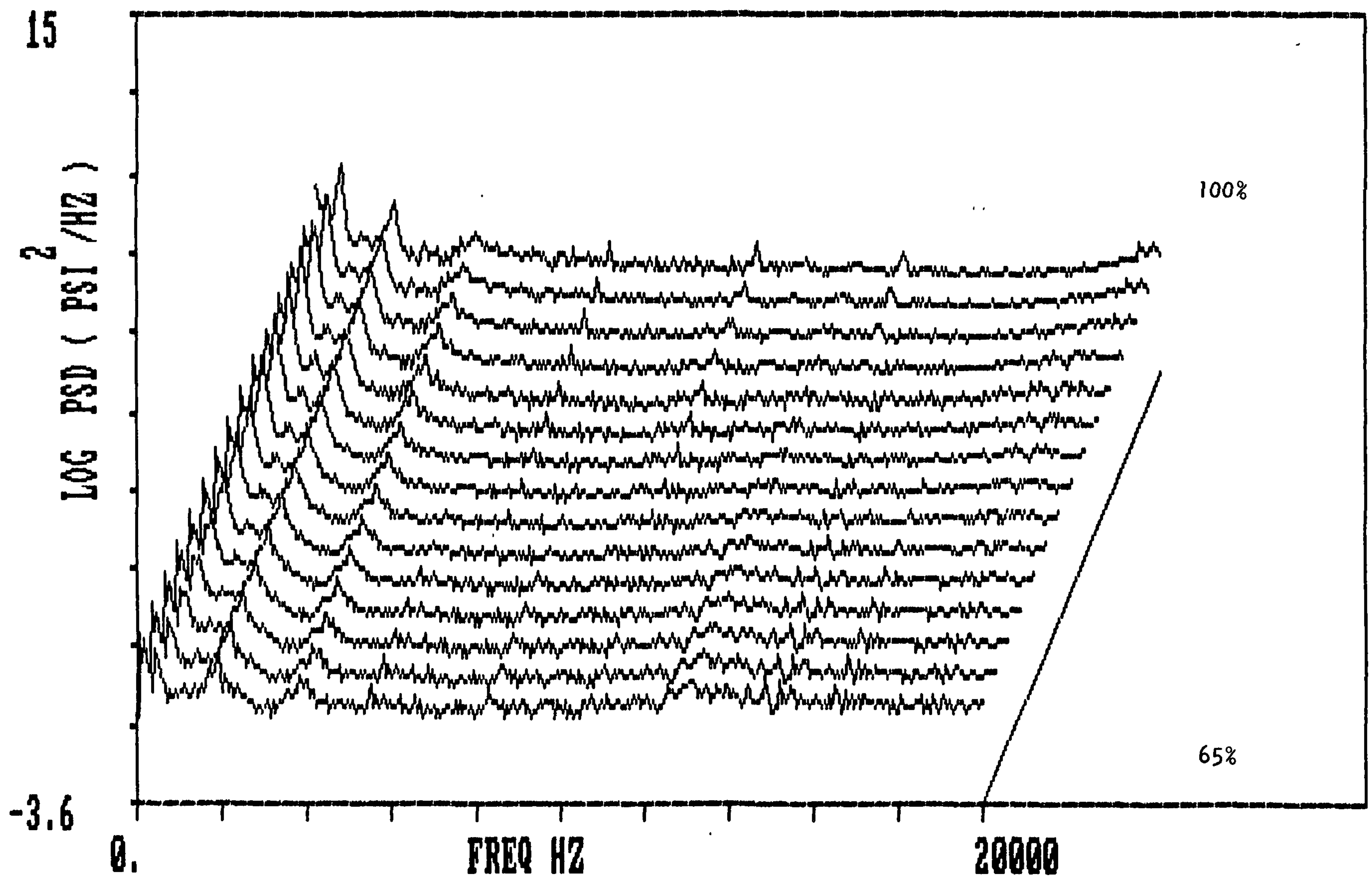


FIGURE 5-21. SPECTRAL TREND HPFP BAL CAV PR (65 - 100% RPL)

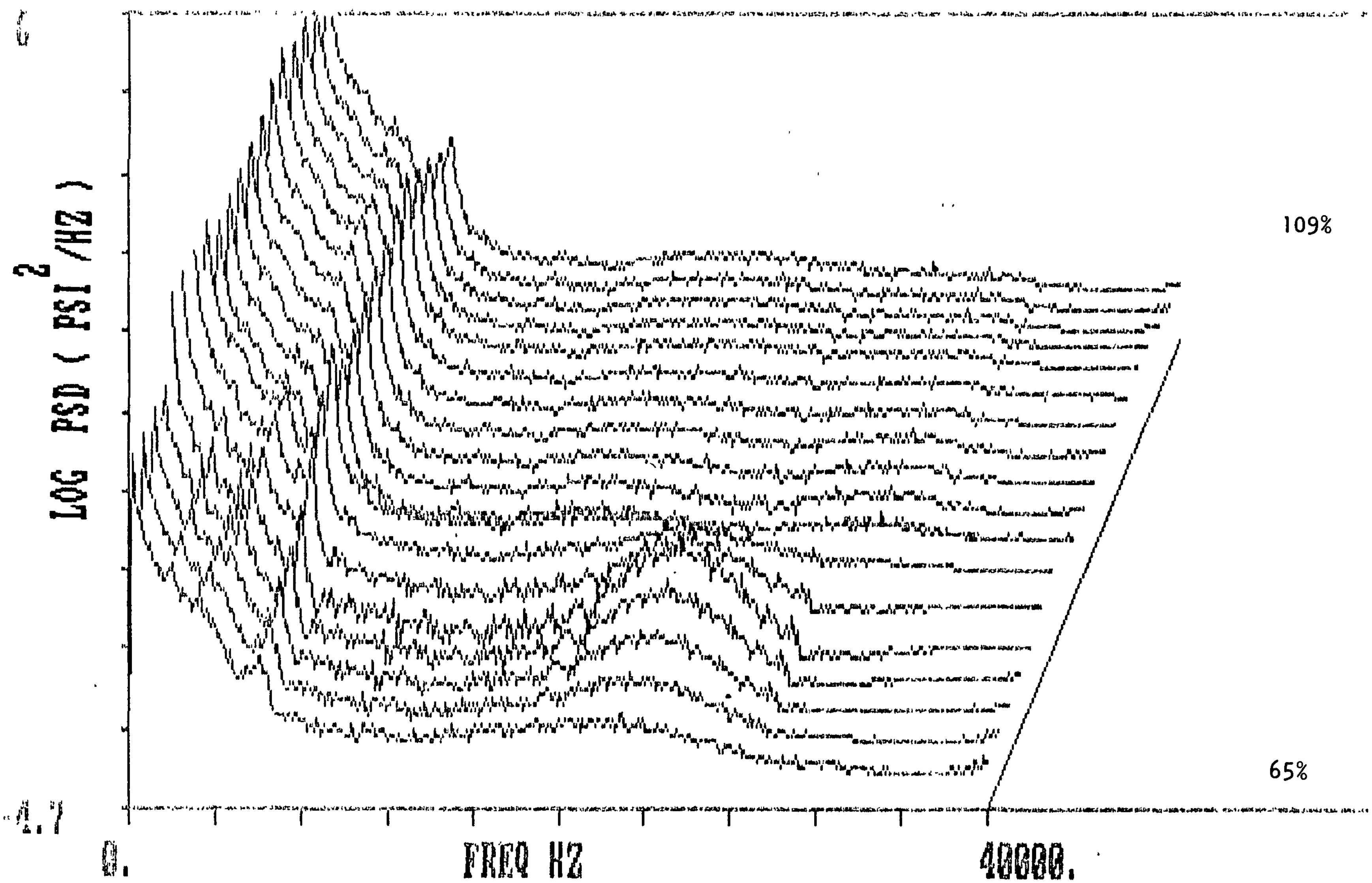


FIGURE 5-22. SPECTRAL TREND MCC FUEL INJ PR (65 - 109% RPL)

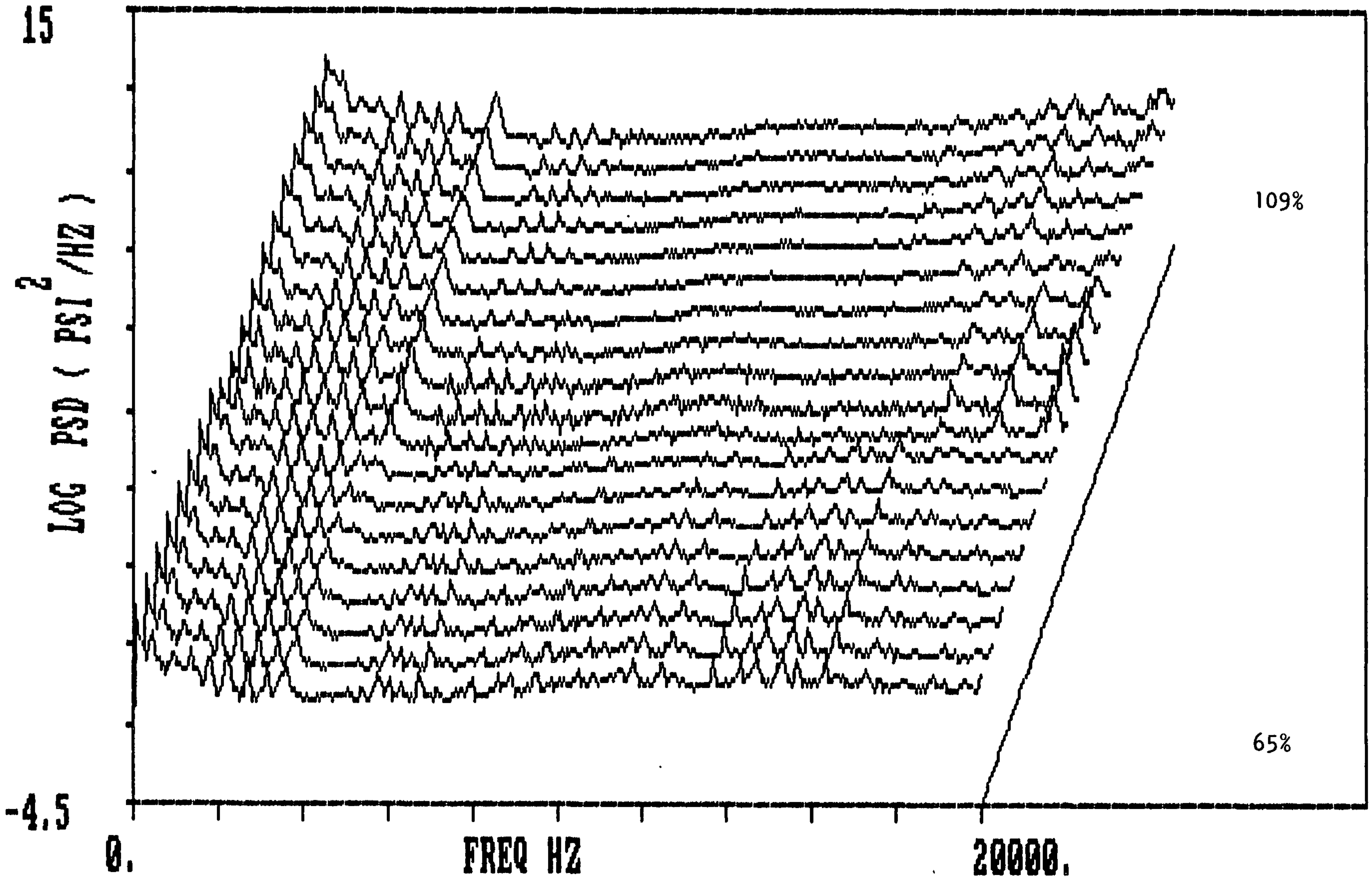


FIGURE 5-23. SPECTRAL TREND FPB PC (65 - 109% RPL)

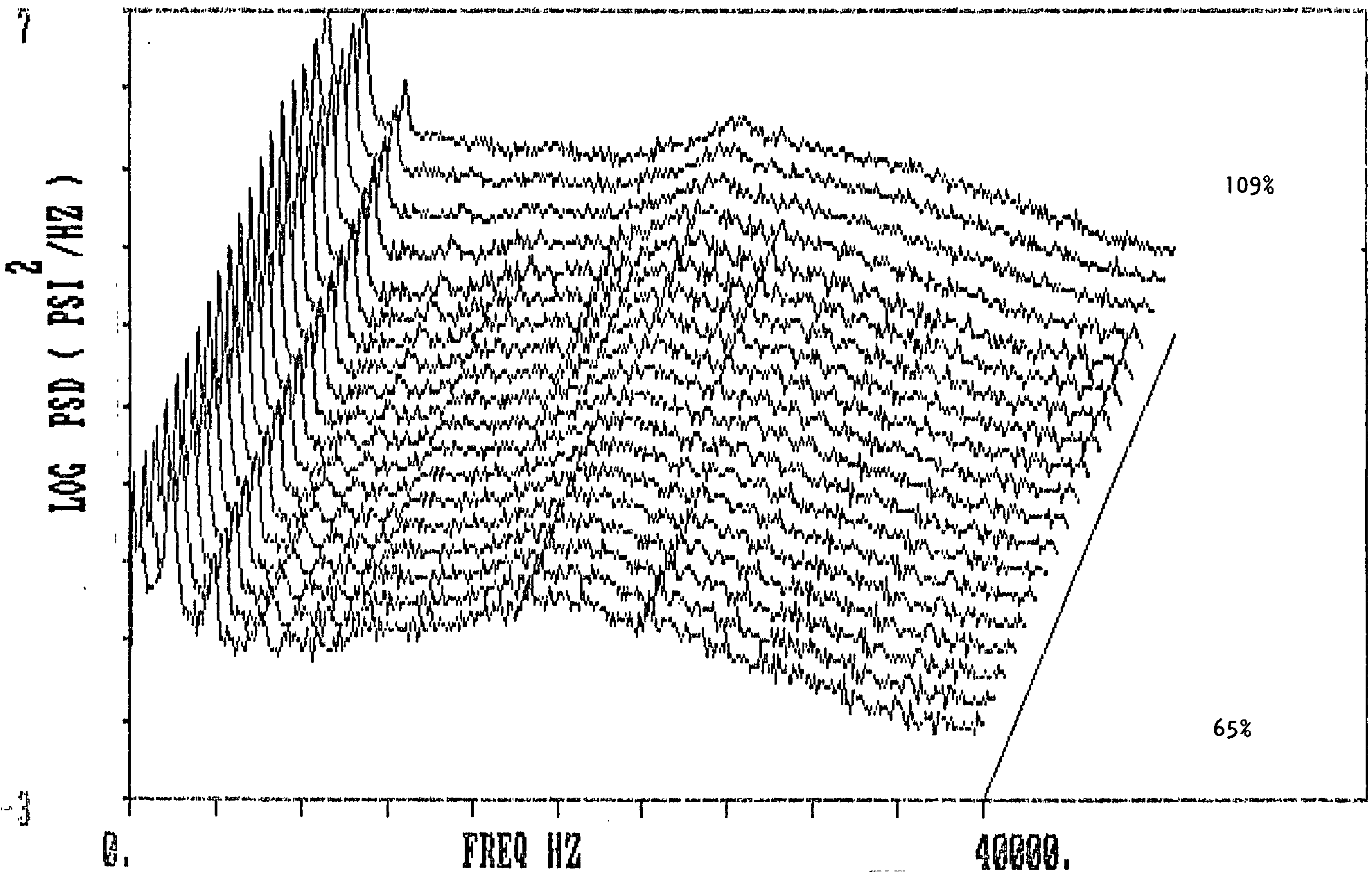


FIGURE 5-24. SPECTRAL TREND MCC FUEL INJ PR DC (65 - 109% RPL)

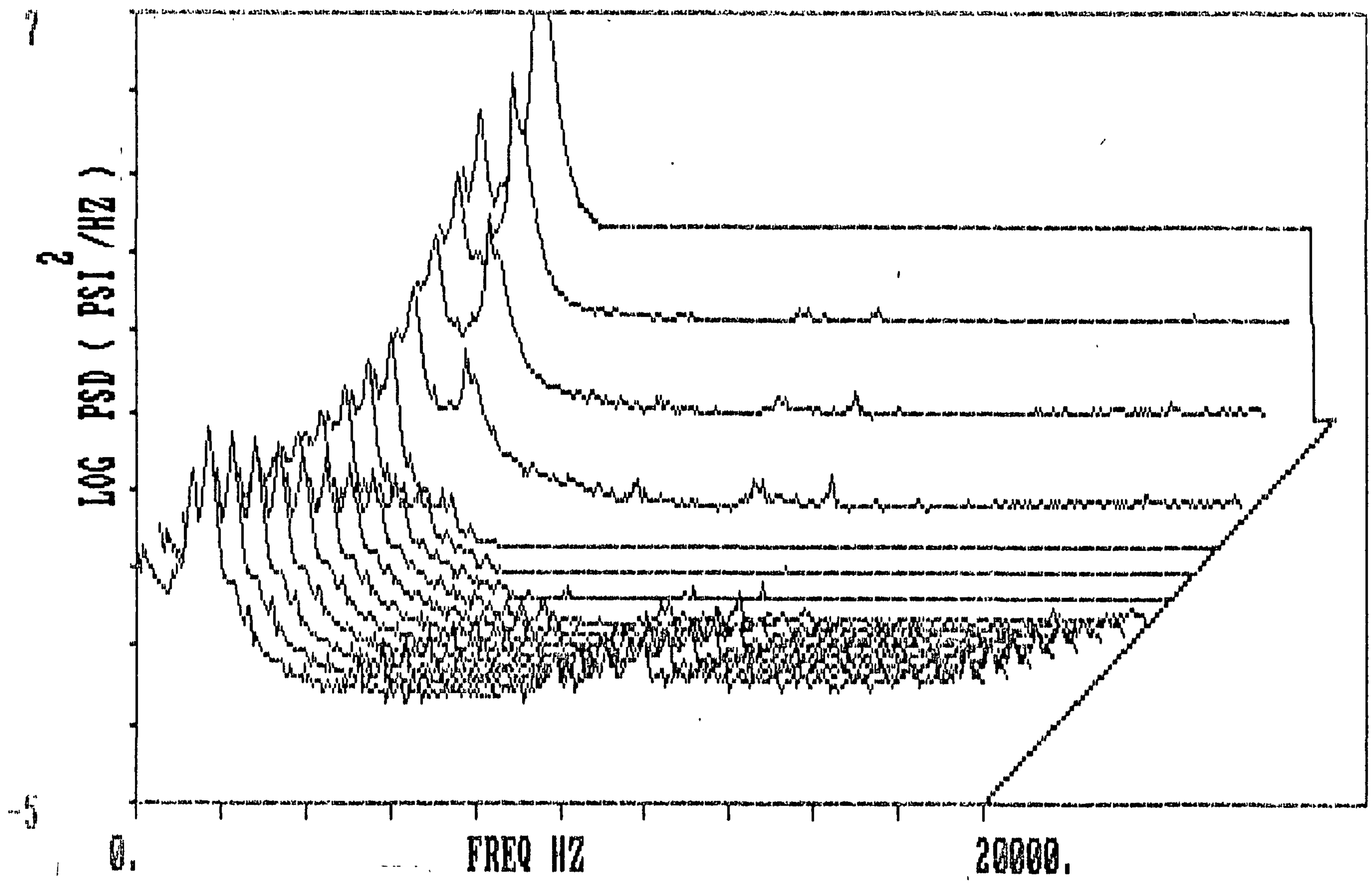


FIGURE 5-25. SPECTRAL TREND HPOP IN PR

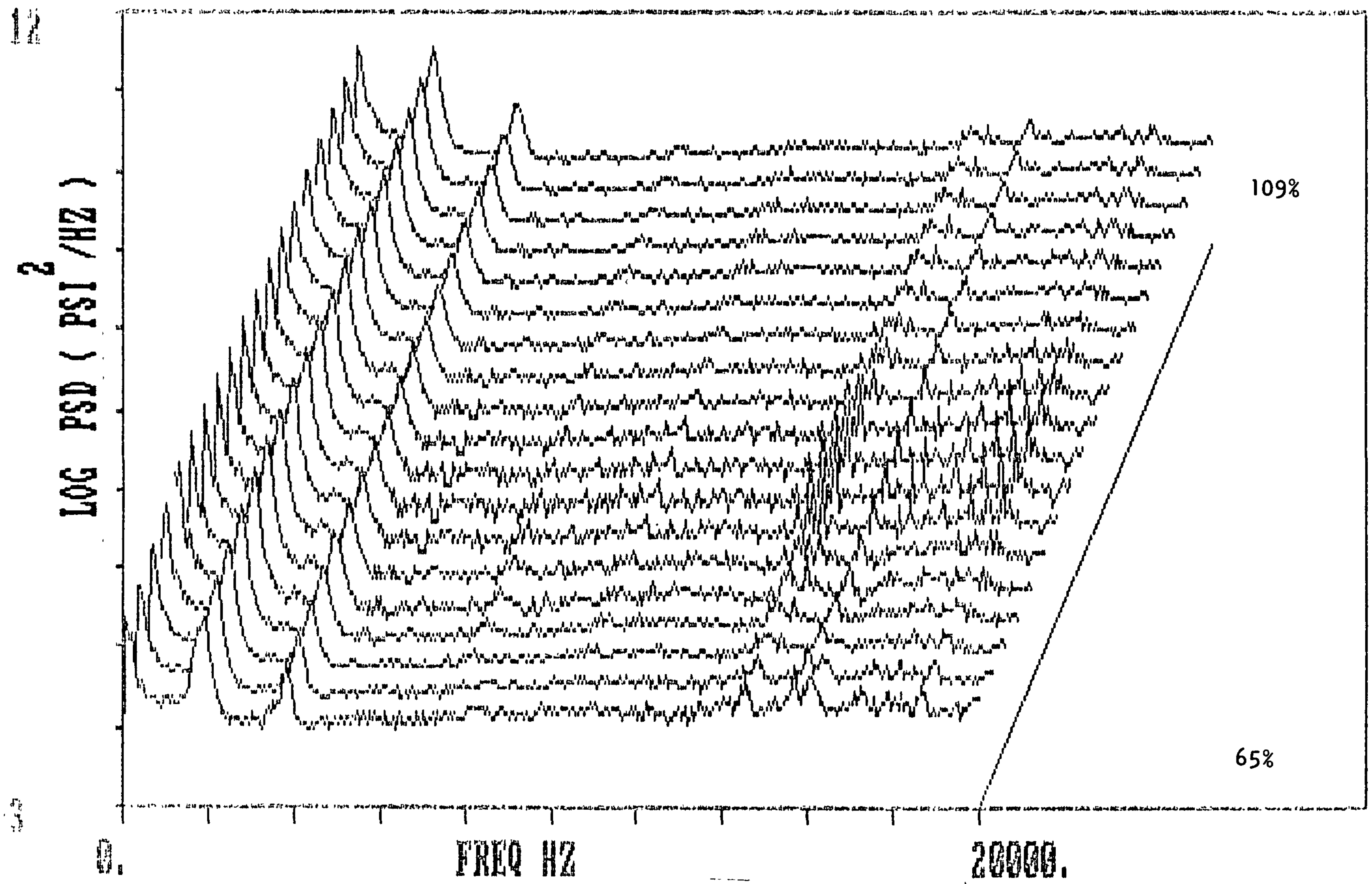


FIGURE 5-26. SPECTRAL TREND OPF PC PR DC (65 - 109% RPL)

TABLE 5-1. A SPECTRUM INTERPOLATION PROGRAM

BASED ON SINGULAR VALUE DECOMPOSITION


```
C       PROGRAM SVDNASA
C       THIS PROGRAM GENERATE THE PSD OF SSME PRESSURE MEASUREMENTS
C       FROM 31 DIFFERENT LOCATIONS.  THE ORIGINAL PSD HAVE BEEN
C       TRANSFORMED TO A FEW BASIC WAVEFORMS BY USING SINGULAR
C       VALUE DECOMPOSITION METHOD.  THESE BASIC WAVE FORMS WERE
C       USED IN THIS PROGRAM TO ESTIMATE THE PSD AT ANY POWER LEVEL
C       WITHIN CERTAIN RANGE.
        REAL*4 XX1(100),XX2(100),WK1(100),WK2(100)
        DIMENSION ID(34),NN(34)
        REAL*4 CC(16,16),DD(16,16),U(400,12),VV(400,12),PSD(400)
        CHARACTER*3 FNAME(34)
        CHARACTER*10 F1,F2,F3
        CHARACTER*11 F11,F12,F13
        CHARACTER*16 F4,F5
        WRITE(*,666)
  666   FORMAT(' THIS PROGRAM GENERATES THE PSD OF PRESSURE MEASURE',
       1 'MENT AT ANY POWER LEVEL',/,' ( WITHIN CERTAIN RANGE ) BY ',
       1 'USING SINGULAR VALUE DECOMPOSITION METHOD')
        WRITE(*,'(A)')' OUTPUT PSD HAS 400 LINE SPECTRA'
        OPEN(1,FILE='SVDNASA.DAT',STATUS='OLD')
        DO 12 I=1,34
        READ(1,601)FNAME(I),ID(I),NN(I)
   12   CONTINUE
  601   FORMAT(A3,1X,I1,1X,I2)
        CLOSE(1)
        M=400
  123   CALL CHANNEL
        WRITE(*,'(A)')' ENTER CHANNEL #'
        READ(*,*)NCH
        IF(ID(NCH).EQ.9)WRITE(*,'(A)')' NO DATA''
        IF(ID(NCH).EQ.9)GOTO 123
CCCCCCCCCCCCCCCCCCCCCCCCCCC ID=2 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        IF(ID(NCH).EQ.2)THEN
        WRITE(*,'(A)')' ENTER 1 FOR MAX FREQ = 20000 HZ '
        IF(NCH.NE.7)WRITE(*,'(A)')'       2 FOR MAX FREQ = 40000 HZ '
        IF(NCH.EQ.7)WRITE(*,'(A)')'       2 FOR MAX FREQ =  5000 HZ'
        READ(*,*)IF12
          IF(IF12.EQ.1)THEN
          WRITE(F11,602)FNAME(NCH)
          WRITE(F12,603)FNAME(NCH)
          WRITE(F13,604)FNAME(NCH)
          ELSE
          WRITE(F11,605)FNAME(NCH)
          WRITE(F12,606)FNAME(NCH)
          WRITE(F13,607)FNAME(NCH)
          ENDIF
        OPEN(1,FILE=F11,STATUS='OLD',ACCESS='DIRECT',RECL=4*M)
        OPEN(2,FILE=F12,STATUS='OLD')
        OPEN(3,FILE=F13,STATUS='OLD')
        WRITE(*,'(A)')F11,F12,F13
        ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCC ID=0,1 CCCCCCCCCCCCCCCCCCCCCCCCCCCC
         IF(ID(NCH).LE.1)THEN
         IF(NCH.NE.31.AND.NCH.NE.30.AND.NCH.NE.9)THEN
         WRITE(*,'(A)')' MAX FREQ. OF OUTPUT PSD = 20000. HZ'
         ELSE
         WRITE(*,'(A)')' MAX FREQ. OF OUTPUT PSD = 40000. HZ'
         ENDIF
         WRITE(F1,608)FNAME(NCH)
```

```
              WRITE(F2,609)FNAME(NCH)
              WRITE(F3,610)FNAME(NCH)
              OPEN(1,FILE=F1,STATUS='OLD',ACCESS='DIRECT',RECL=4*M)
              IF(ID(NCH).EQ.1)OPEN(2,FILE=F2,STATUS='OLD')
              OPEN(3,FILE=F3,STATUS='OLD')
              ENDIF
     602     FORMAT('PSD',A3,'A.EVT')
     603     FORMAT('PSD',A3,'A.EVL')
     604     FORMAT('PSD',A3,'A.PRO')
     605     FORMAT('PSD',A3,'B.EVT')
     606     FORMAT('PSD',A3,'B.EVL')
     607     FORMAT('PSD',A3,'B.PRO')
     608     FORMAT('PSD',A3,'.EVT')
     609     FORMAT('PSD',A3,'.EVL')
     610     FORMAT('PSD',A3,'.PRO')
             M=400
             N=NN(NCH)
              READ(3,*)(XX1(I),I=2,N+1)
              CLOSE(3)
              DO 22 J=1,N
1             READ(1,REC=J)(U(I,J),I=1,M)
1    22       CONTINUE
              DO 1 J=1,N
1             IF(ID(NCH).NE.0)READ(2,*)(CC(I,J),I=1,N)
1    1        CONTINUE
              CLOSE(1)
              IF(ID(NCH).NE.0)CLOSE(2)
     321     NOUT=1
             WRITE(*,'(A)')' ENTER FILE NAME OF OUTPUT PSD'
             READ(*,'(A)')F5
             OPEN(5,FILE=F5,STATUS='NEW')
     CCCCCCCCCCCCCCCCCCCCCCC  ID=0  CCCCCCCCCCCCCCCCCCCCCCCCCC
             IF(ID(NCH).EQ.0)THEN
     621     FORMAT(' PSD ONLY AVAILABLE AT POWER LEVEL: ',10(F7.2,'% '))
             WRITE(*,621)(XX1(I),I=2,N+1)
             WRITE(*,'(A)')' ENTER DESIRED POWER (IN %)'
             READ(*,*)PWL
              AMIN=200
              MIN=2
             DO 691 I=2,N+1
1            TE=ABS(PWL-XX1(I))
1            IF(TE.LE.AMIN)THEN
1            AMIN=TE
1            MIN=1-1
1            ENDIF
1    691     CONTINUE
             WRITE(5,*)(U(I,MIN),1=1,M)
             CLOSE(5)
             GOTO 1986
             ENDIF
     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
             IF(ID(NCH).EQ.1)THEN
             WRITE(*,611)XX1(2),XX1(N+1)
     611     FORMAT(' AVAILABLE POWER LEVEL = ',F10.4,'% TO ',F10.4,'%')
             WRITE(*,'(A)')' ENTER DESIRED POWER LEVEL (%) '
             READ(*,*)XX2(1)
             ELSE
              WRITE(*,'(A)')' ENTER INPUT POWER LEVEL FILNAM'
              READ(*,'(A)')F4
```

```
D Line# 1      7                                    Microsoft FORTRAN77 V3.20 02/84
                OPEN(4,FILE=F4,STATUS='OLD')
                READ(4,*)(XX2(I),I=1,NOUT)
                CLOSE(4)
               ENDIF
               IFLAR=2
               WRITE(*,612)N
               READ(*,*)MMM
         612   FORMAT(' ENTER # OF BASIC WAVEFORMS TO BE USED (LE.',I3,')')
               IFL=1
               XX1(1)=XX1(2)-(XX1(3)-XX1(2))
               XX1(N+2)=XX1(N+1)+(XX1(N+1)-XX1(N))
               DO 11 I=1,M
1              DO 11 J=1,MMM
2              VV(I,J)=U(I,J)
2        11    CONTINUE
               DO 66 I=1,MMM
1              DO 65 J=1,N
2        65    WK1(J+1)=CC(I,J)
1              WK1(1)=WK1(2)+(WK1(2)-WK1(3))
1              WK1(N+2)=WK1(N+1)+(WK1(N+1)-WK1(N))
1              IF(IFLAR.EQ.1)THEN
1              CALL LAR(XX1,WK1,N+2,NOUT,XX2,WK2)
1              ELSE
1              CALL LINE(XX1,WK1,N+2,NOUT,XX2,WK2)
1              ENDIF
1              DO 64 J=1,NOUT
2        64    DD(I,J)=WK2(J)
1        66    CONTINUE
         C**********************************
               DO 102 II=1,NOUT
1              DO 101 I=1,M
2              PSD(I)=0.
2              DO 101 K=1,MMM
3              PSD(I)=PSD(I)+VV(I,K)*DD(K,II)
3        101   CONTINUE
1               IF(IFL.EQ.1)THEN
1               DO 104 LL=1,M
2        104    PSD(LL)=10.**(PSD(LL))
1               ENDIF
1              WRITE(5,103)(PSD(I),I=1,M)
1        102   CONTINUE
         103   FORMAT(1X,5E15.7)
               CLOSE(5)
         1986  WRITE(*,'(A)')' ENTER 1 FOR SELECTING ANOTHER POWER LEVEL'
               WRITE(*,'(A)')'       2 FOR SELECTING ANOTHER CHANNEL'
               WRITE(*,'(A)')'       ELSE FOR STOP PROGROM'
               READ(*,*)IFCG
               IF(IFCG.EQ.2)GOTO 123
               IF(IFCG.EQ.1)GOTO 321
               STOP
               END
```

Name	Type	Offset	P	Class
ABS				INTRINSIC
AMIN	REAL	44566		
CC	REAL	41874		
DD	REAL	42898		
F1	CHAR*10	44261		

D Line# 1 7 Microsoft FORTRAN77 V3.20 02/84

F11	CHAR*11	44228
F12	CHAR*11	44239
F13	CHAR*11	44250
F2	CHAR*10	44271
F3	CHAR*10	44281
F4	CHAR*16	44636
F5	CHAR*16	44490
FNAME	CHAR*3	43922
I	INTEGER*4	44196
ID	INTEGER*4	1602
IF12	INTEGER*4	44224
IFCG	INTEGER*4	44780
IFL	INTEGER*4	44716
IFLAR	INTEGER*4	44652
II	INTEGER*4	44740
J	INTEGER*4	44474
K	INTEGER*4	44752
LL	INTEGER*4	44760
M	INTEGER*4	44216
MIN	INTEGER*4	44570
MMM	INTEGER*4	44656
N	INTEGER*4	44466
NCH	INTEGER*4	44220
NN	INTEGER*4	1738
NOUT	INTEGER*4	44486
PSD	REAL	40274
PWL	REAL	44562
TE	REAL	44578
U	REAL	1874
VV	REAL	21074
WK1	REAL	802
WK2	REAL	1202
XX1	REAL	2
XX2	REAL	402

```
      C***************************************************
                SUBROUTINE LAR(XX1,YY1,NIN,NOUT,XX2,YY2)
                REAL*4 XX1(1),YY1(1),XX2(1),YY2(1),CC(50),FF(50)
                DO 3 I=1,NIN
1               CC(I)=1.00
1               DO 4 J=1,NIN
2               IF(J.EQ.I)GOTO 4
2               CC(I)=CC(I)*(XX1(I)-XX1(J))
2         4     CONTINUE
1               CC(I)=YY1(I)/CC(I)
1         3     CONTINUE
                DO 5 I=1,NOUT
1               A=XX2(I)
1                DO 6 J=1,NIN
2                FF(J)=1.00
2                DO 7 K=1,NIN
3                IF(K.EQ.J)GOTO 7
3                FF(J)=FF(J)*(A-XX1(K))
3         7      CONTINUE
2                FF(J)=FF(J)*CC(J)
2         6      CONTINUE
1               YY2(I)=0.0
1                DO 8 L=1,NIN
```

```
D Line# 1      7
2    193        YY2(I)=YY2(I)+FF(L)
2    194 8      CONTINUE
1    195 5     CONTINUE
     196       RETURN
     197       END

Name     Type          Offset P Class

A        REAL           45204
CC       REAL           44784
FF       REAL           44984
I        INTEGER*4      45184
J        INTEGER*4      45192
K        INTEGER*4      45212
L        INTEGER*4      45220
NIN      INTEGER*4          8 *
NOUT     INTEGER*4         12 *
XX1      REAL               0 *
XX2      REAL              16 *
YY1      REAL               4 *
YY2      REAL              20 *


     198       SUBROUTINE LINE(XX1,YY1,NIN,NOUT,XX2,YY2)
     199       REAL*4 XX1(1),YY1(1),XX2(1),YY2(1)
     200       DO 99 I=1,NOUT
1    201       DO 1 J=1,NIN-1
2    202       IF(XX2(I).GE.XX1(J).AND.XX2(I).LE.XX1(J+1))THEN
2    203       YY2(I)=SSS(XX1(J),XX1(J+1),YY1(J),YY1(J+1),XX2(I))
2    204       GOTO 99
2    205       ENDIF
2    206 1     CONTINUE
1    207 99    CONTINUE
     208       RETURN
     209       END

Name     Type          Offset P Class

I        INTEGER*4      45228
J        INTEGER*4      45236
NIN      INTEGER*4          8 *
NOUT     INTEGER*4         12 *
XX1      REAL               0 *
XX2      REAL              16 *
YY1      REAL               4 *
YY2      REAL              20 *


     210       FUNCTION SSS(X1,X2,Y1,Y2,XX)
     211       SSS=Y1+(XX-X1)*(Y2-Y1)/(X2-X1)
     212       RETURN
     213       END

Name     Type          Offset P Class

X1       REAL               0 *
X2       REAL               4 *
XX       REAL              16 *
Y1       REAL               8 *
```

```
D Line# 1      7                              Microsoft FORTRAN77 V3.20 02/84
Y2      REAL              12 *
```

```
      SUBROUTINE CHANNEL
      WRITE(*,'(A)')' 1.  THRUST(NO DATA)       12. LPFP DS PR
     1 23. FPB PC'
      WRITE(*,'(A)')' 2.  LPFP IN PR            13. PBP FUEL SUP PR
     1 24. MCC FUEL INJ PR DC'
      WRITE(*,'(A)')' 3.  LPOP DS PR            14. FPB FUEL MAN PR
     1 25. OPB PL*'
      WRITE(*,'(A)')' 4.  HPOP BAL CAV PR2      15. LPOT TURB DR PR
     1 26. HPOP IN PR'
      WRITE(*,'(A)')' 5.  HPOP BAL CAV PR1      16. LOX HX OUT PR
     1 27. AUX LX I PR*'
      WRITE(*,'(A)')' 6.  HPOP DS PR            17. OPB PC PRR*
     1 28. HI LX I PR*'
      WRITE(*,'(A)')' 7.  LOX IN DUCT PR2       18. HEX OUTLET*
     1 29. HOT GAS MAN PR2*'
      WRITE(*,'(A)')' 8.  HPOP DISC PR          19. HPFP DS PR*
     1 30. MCC LONG ACCEL 1*'
      WRITE(*,'(A)')' 9.  MCC HOT GAS IN PR     20. HOT GAS MAN PR3*
     1 31. MCC IN PR*'
      WRITE(*,'(A)')' 10. PBP DS PR             21. HPFP BAL CAV PR
     1 34. OPB PC PR DC'
      WRITE(*,'(A)')' 11. CHAMBER PR*           22. MCC FUEL INJ PR '
      WRITE(*,666)
      WRITE(*,667)
666   FORMAT(' NOTE: CHANNEL DESCRIPTOR ENDED WITH "*" MEANS',
     1  ' SVD DATA IS NOT AVAILABLE')
667    FORMAT('        PSD ONLY AVAILABLE AT SOME FIXED POWER LEVEL')
      RETURN
      END
```

Name	Type	Offset	P	Class

Name	Type	Size	Class
CHANNE			SUBROUTINE
LAR			SUBROUTINE
LINE			SUBROUTINE
MAIN			PROGRAM
SSS	REAL		FUNCTION

Pass One   No Errors Detected
242 Source Lines

```
D Line# 1      7                                   Microsoft FORTRAN77 V3.20 02/84
        C      THIS PROGRAM USES SINGULATR VALUE DECOMPOSITION METHOD TO
        C      GENERATE THE BASIC WAVEFORMS AND ITS SINGULAR VALUES IN ORDER
        C      TO ESTABISH THE WHOLE FAMILY OF PSD FROM THE GIVEN MEASURED
        C      PSD AT SOME CERTAIN PARAMETER ( e.g. POWER LEVEL )
               DIMENSION IJK(12),WK(12)
               REAL*4 CC(11,11),DD(11,11)
        C      REAL*4 A(M,N),U(M,N),V(M,N),SIGMA(M),WORK(M)
        C      INPUT # OF DATA IN PSD MUST EQUAL M
        C      INPUT # OF PSD COULD BE SMALL THEN N
               REAL*4 A(400,11),U(400,11),V(400,11),SIGMA(400),WORK(400)
               CHARACTER*16 F1,F2,F3
               WRITE(*,'(A)')' ENTER # OF DATA IN PSD ( MUST EQUAL 400)'
               READ(*,*)M
               WRITE(*,'(A)')' ENTER # OF PSD (LE. 11)'
               READ(*,*)N
               WRITE(*,'(A)')' ENTER INPUT FILNAM'
               READ(*,'(A)')F1
               WRITE(*,'(A)')' ENTER OUTPUT BASIC WAVEFORM FILNAM'
               READ(*,'(A)')F2
               WRITE(*,'(A)')' ENTER OUTPUT SVD COEFFICIENT FILNAM'
               READ(*,'(A)')F3
               WRITE(*,'(A)')' ENTER 1 FOR SORTING SV'
               READ(*,*)IFST
               WRITE(*,'(A)')' ENTER 1 FOR LOG Y'
               READ(*,*)LOGY
               OPEN(1,FILE=F1,STATUS='OLD')
               OPEN(2,FILE=F2,STATUS='NEW')
               OPEN(3,FILE=F3,STATUS='NEW')
               DO 1 I=1,N
1          1   READ(1,*)(A(J,I),J=1,M)
               IF(LOGY.NE.1)GOTO 79
               DO 78 I=1,N
1              DO 78 J=1,M
2          78  A(J,I)=ALOG10(A(J,I))
           79  CONTINUE
               NM=M
               CALL SVD(NM,M,N,A,SIGMA,.TRUE.,U,.TRUE.,V,IERR,WORK)
               WRITE(*,'(A)')' VECTOR NORM OF U ARE:'
               DO 501 J=1,N
1              P=0.
1              DO 502 I=1,M
2          502 P=P+U(I,J)**2
1              WRITE(*,*)P
1          501 CONTINUE
               WRITE(*,'(A)')' VECTOR NORM FOR V ARE:'
               DO 503 J=1,N
1              P=0.
1              DO 504 I=1,N
2          504 P=P+V(I,J)**2
1              WRITE(*,*)P
1          503 CONTINUE
               WRITE(*,'(A)')' SINGULAR VALUES ARE:'
               IF(IFST.EQ.1)THEN
               DO 777 I=1,N
1          777 WK(I)=SIGMA(I)
               CALL SORT(WK,IJK,N)
               ELSE
               DO 333 I=1,N
1          333 IJK(I)=I
```

```
D Line# 1      7                        Microsoft FORTRAN77 V3.20 02/84
               ENDIF
          C    WRITE(*,'(A)')' SORTING ORDER OF SINGULER VALUES:'
               WRITE(*,*)(SIGMA(IJK(I)),I=1,N)
               IF(IERR.NE.0)WRITE(*,2)IERR
          2    FORMAT('    TROUBLE.   IERR=',I4)
               DO 3 I=1,N
1              DO 3 J=1,N
2         3    CC(I,J)=V(I,J)*SIGMA(J)
               DO 666 I=1,N
1              DO 666 J=1,N
2         666  DD(I,J)=CC(J,I)
               DO 201 I=1,N
1              DO 201 J=1,M
2              A(J,I)=U(J,IJK(I))
2         201  CONTINUE
               DO 202 I=1,N
1              DO 202 J=1,N
2         202  CC(I,J)=DD(IJK(I),J)
               DO 4 J=1,N
1              WRITE(2,*)(A(I,J),I=1,M)
1              WRITE(3,*)(CC(I,J),I=1,N)
1         4    CONTINUE
               WRITE(3,*)(SIGMA(IJK(I)),I=1,N)
               STOP
               END
```

Name	Type	Offset	P	Class
A	REAL	39466		
ALOG10				INTRINSIC
CC	REAL	38498		
DD	REAL	38982		
F1	CHAR*16	57074		
F2	CHAR*16	57090		
F3	CHAR*16	57106		
I	INTEGER*4	57130		
IERR	INTEGER*4	57158		
IFST	INTEGER*4	57122		
IJK	INTEGER*4	2		
J	INTEGER*4	57138		
LOGY	INTEGER*4	57126		
M	INTEGER*4	57066		
N	INTEGER*4	57070		
NM	INTEGER*4	57154		
P	REAL	57166		
SIGMA	REAL	36898		
U	REAL	1698		
V	REAL	19298		
WK	REAL	50		
WORK	REAL	98		

```
               SUBROUTINE SORT(X,IJK,N)
               DIMENSION X(1),IJK(1)
               AMIN=X(1)
               DO 1 I=1,N
1              IF(X(I).LT.AMIN)AMIN=X(I)
1         1    CONTINUE
               AMIN=AMIN-100.
```

```
D Line# 1       7
                DO 2 I=1,N
1               CALL FINDMAX(X,N,MAX)
1               IJK(I)=MAX
1               X(MAX)=AMIN
1         2     CONTINUE
                RETURN
                END
```

Name	Type	Offset	P	Class
AMIN	REAL	57254		
I	INTEGER*4	57258		
IJK	INTEGER*4	4	*	
MAX	INTEGER*4	57270		
N	INTEGER*4	8	*	
X	REAL	0	*	

```
          C******************************
                SUBROUTINE FINDMAX(X,N,MAX)
                REAL*4 X(1)
                AMAX=X(1)
                MAX=1
                DO 1 I=2,N
1               IF(X(I).GT.AMAX)THEN
1               AMAX=X(I)
1               MAX=I
1               ENDIF
1         1     CONTINUE
                RETURN
                END
```

Name	Type	Offset	P	Class
AMAX	REAL	57274		
I	INTEGER*4	57278		
MAX	INTEGER*4	8	*	
N	INTEGER*4	4	*	
X	REAL	0	*	

```
                SUBROUTINE SVD(NM,M,N,A,W,MATU,U,MATV,V,IERR,RV1)
                REAL*4 A(NM,N),W(N),U(NM,N),V(NM,N),RV1(N)
                LOGICAL MATU,MATV
                IERR=0
                DO 100 I=1,M
1               DO 100 J=1,N
2               U(I,J)=A(I,J)
2         100   CONTINUE
                G=0.
                SCALE=0.
                ANORM=0.
                DO 300 I=1,N
1               L=I+1
1               RV1(I)=SCALE*G
1               G=0.
1               S=0
1               SCALE=0.
1               IF(I.GT.M)GOTO 210
```

```
1          DO 120 K=I,M
2     120  SCALE=SCALE+ABS(U(K,I))
1          IF(SCALE.EQ.0.0)GOTO 210
1          DO 130 K=I,M
2          U(K,I)=U(K,I)/SCALE
2          S=S+U(K,I)**2
2     130  CONTINUE
1          F=U(I,I)
1          G=-SIGN(SQRT(S),F)
1          H=F*G-S
1          U(I,I)=F-G
1          IF(I.EQ.N)GOTO 190
1          DO 150 J=L,N
2          S=0.
2          DO 140 K=I,M
3     140  S=S+U(K,I)*U(K,J)
2          F=S/H
2          DO 150 K=I,M
3          U(K,J)=U(K,J)+F*U(K,I)
3     150  CONTINUE
1     190  DO 200 K=I,M
2     200  U(K,I)=SCALE*U(K,I)
1     210  W(I)=SCALE*G
1          G=0.
1          S=0.
1          SCALE=0.
1          IF(I.GT.M.OR.I.EQ.N)GOTO 290
1          DO 220 K=L,N
2     220  SCALE=SCALE+ABS(U(I,K))
1          IF(SCALE.EQ.0.0)GOTO 290
1          DO 230 K=L,N
2          U(I,K)=U(I,K)/SCALE
2          S=S+U(I,K)**2
2     230  CONTINUE
1          F=U(I,L)
1          G=-SIGN(SQRT(S),F)
1          H=F*G-S
1          U(I,L)=F-G
1          DO 240 K=L,N
2     240  RV1(K)=U(I,K)/H
1          IF(I.EQ.M)GOTO 270
1          DO 260 J=L,M
2          S=0.
2          DO 250 K=L,N
3     250  S=S+U(J,K)*U(I,K)
2          DO 260 K=L,N
3          U(J,K)=U(J,K)+S*RV1(K)
3     260  CONTINUE
1     270  DO 280 K=L,N
2     280  U(I,K)=SCALE*U(I,K)
1     290  ANORM=AMAX1(ANORM,ABS(W(I))+ABS(RV1(I)))
1     300  CONTINUE
           IF(.NOT.MATV)GOTO 410
           DO 400 II=1,N
1          I=N+1-II
1          IF(I.EQ.N)GOTO 390
1          IF(G.EQ.0.0)GOTO 360
1          DO 320 J=L,N
2     320  V(J,I)=(U(I,J)/U(I,L))/G
```

```
D Line# 1      7
1              DO 350 J=L,N
2              S=0.
2              DO 340 K=L,N
3        340   S=S+U(I,K)*V(K,J)
2              DO 350 K=L,N
3              V(K,J)=V(K,J)+S*V(K,I)
3        350   CONTINUE
1        360   DO 380 J=L,N
2              V(I,J)=0.
2              V(J,I)=0.
2        380   CONTINUE
1        390   V(I,I)=1.
1              G=RV1(I)
1              L=I
1        400   CONTINUE
         410   IF(.NOT.MATU)GOTO 510
               MN=N
               IF(M.LT.N)MN=M
               DO 500 II=1,MN
1              I=MN+1-II
1              L=I+1
1              G=W(I)
1              IF(I.EQ.N)GOTO 430
1              DO 420 J=L,N
2        420   U(I,J)=0.
1        430   IF(G.EQ.0.)GOTO 475
1              IF(I.EQ.MN)GOTO 460
1              DO 450 J=L,N
2              S=0.
2              DO 440 K=L,M
3        440   S=S+U(K,I)*U(K,J)
2              F=(S/U(I,I))/G
2              DO 450 K=I,M
3              U(K,J)=U(K,J)+F*U(K,I)
3        450   CONTINUE
1        460   DO 470 J=I,M
2        470   U(J,I)=U(J,I)/G
1              GOTO 490
1        475   DO 480 J=I,M
2        480   U(J,I)=0.
1        490   U(I,I)=U(I,I)+1.
1        500   CONTINUE
         510   DO 700 KK=1,N
1              K1=N-KK
1              K=K1+1
1              ITS=0
1        520   DO 530 LL=1,K
2              L1=K-LL
2              L=L1+1
2              IF(ABS(RV1(L))+ANORM.EQ.ANORM)GOTO 565
2              IF(ABS(W(L1))+ANORM.EQ.ANORM)GOTO 540
2        530   CONTINUE
1        540   C=0
1              S=1.
1              DO 560 I=L,K
2              F=S*RV1(I)
2              RV1(I)=C*RV1(I)
2              IF(ABS(F)+ANORM.EQ.ANORM)GOTO 565
2              G=W(I)
```

```
D Line# 1  7
2          H=SQRT(F*F+G*G)
2          W(I)=H
2          C=G/H
2          S=-F/H
2          IF(.NOT.MATU)GOTO 560
2          DO 550 J=1,M
3          Y=U(J,L1)
3          Z=U(J,I)
3          U(J,L1)=Y*C+Z*S
3          U(J,I)=-Y*S+Z*C
3    550   CONTINUE
2    560   CONTINUE
1    565   Z=W(K)
1          IF(L.EQ.K)GOTO 650
1          IF(ITS.EQ.30)GOTO 1000
1          ITS=ITS+1
1          X=W(L)
1          Y=W(K1)
1          G=RV1(K1)
1          H=RV1(K)
1          F=((Y-Z)*(Y+Z)+(G-H)*(G+H))/(2.*H*Y)
1          G=SQRT(F*F+1.)
1          F=((X-Z)*(X+Z)+H*(Y/(F+SIGN(G,F))-H))/X
1          C=1.
1          S=1.
1          DO 600 I1=L,K1
2          I=I1+1
2          G=RV1(I)
2          Y=W(I)
2          H=S*G
2          G=C*G
2          Z=SQRT(F*F+H*H)
2          RV1(I1)=Z
2          C=F/Z
2          S=H/Z
2          F=X*C+G*S
2          G=-X*S+G*C
2          H=Y*S
2          Y=Y*C
2          IF(.NOT.MATV)GOTO 575
2          DO 570 J=1,N
3          X=V(J,I1)
3          Z=V(J,I)
3          V(J,I1)=X*C+Z*S
3          V(J,I)=-X*S+Z*C
3    570   CONTINUE
2    575   Z=SQRT(F*F+H*H)
2          W(I1)=Z
2          IF(Z.EQ.0.0)GOTO 580
2          C=F/Z
2          S=H/Z
2    580   F=C*G+S*Y
2          X=-S*G+C*Y
2          IF(.NOT.MATU)GOTO 600
2          DO 590 J=1,M
3          Y=U(J,I1)
3          Z=U(J,I)
3          U(J,I1)=Y*C+Z*S
3          U(J,I)=-Y*S+Z*C
```

```
D Line# 1       7
3       590     CONTINUE
2       600     CONTINUE
1               RV1(L)=0.
1               RV1(K)=F
1               W(K)=X
1               GOTO 520
1       650     IF(Z.GE.0.0)GOTO 700
1               W(K)=-Z
1               IF(.NOT.MATV)GOTO 700
1               DO 690 J=1,N
2       690     V(J,K)=-V(J,K)
1       700     CONTINUE
                GOTO 1001
        1000    IERR=K
        1001    RETURN
                END
```

Name	Type	Offset	P	Class
A	REAL	12	*	
ABS				INTRINSIC
AMAX1				INTRINSIC
ANORM	REAL	57310		
C	REAL	57478		
F	REAL	57338		
G	REAL	57302		
H	REAL	57342		
I	INTEGER*4	57286		
I1	INTEGER*4	57502		
IERR	INTEGER*4	36	*	
II	INTEGER*4	57390		
ITS	INTEGER*4	57462		
J	INTEGER*4	57294		
K	INTEGER*4	57326		
K1	INTEGER*4	57458		
KK	INTEGER*4	57450		
L	INTEGER*4	57318		
L1	INTEGER*4	57474		
LL	INTEGER*4	57466		
M	INTEGER*4	4	*	
MATU	LOGICAL*4	20	*	
MATV	LOGICAL*4	28	*	
MN	INTEGER*4	57418		
N	INTEGER*4	8	*	
NM	INTEGER*4	0	*	
RV1	REAL	40	*	
S	REAL	57322		
SCALE	REAL	57306		
SIGN				INTRINSIC
SQRT				INTRINSIC
U	REAL	24	*	
V	REAL	32	*	
W	REAL	16	*	
X	REAL	57498		
Y	REAL	57490		
Z	REAL	57494		

APPENDIX A

FLOW DYNAMIC ENVIRONMENT DATA BASE DEVELOPMENT FOR THE SSME

C. V. Sundaram
Wyle Laboratories, Huntsville

Abstract

This paper describes the studies being carried out on the fluid flow-induced vibration of the Space Shuttle main engine (SSME) components. This study is being carried out with a view to correlating the frequency characteristics of the pressure fluctuations in a rocket engine to its operating conditions and geometry. An overview of the data base development for SSME test firing results and the interactive computer software used to access, retrieve, and plot or print the results selectively for given thrust levels, engine numbers, etc., is presented. The various statistical methods available in the computer code for data analysis are discussed. Plots of test data, nondimensionalized using parameters such as fluid flow velocities, densities, and pressures, are presented. Comparative studies of these results with results available in the literature are made. Correlations between the resonant peaks observed at higher frequencies in power spectral density plots with pump geometry and operating conditions are discussed. Finally, an overview of the status of the investigation is presented and future directions are discussed.

Introduction

The development efforts expended on the main engine of the Space Shuttle has resulted in the evolution of highly complex machinery which is capable of generating high thrust levels. Even from the earliest days of spacecraft engine development, achieving the highest possible specific thrust, based on engine weight or overall size, has been identified as one of the prime goals. The earlier spacecraft, however, were equipped with power sources which were designed for just one mission. The advent of the concept of reusable space vehicles added a new dimension to the

engine design, that they not only satisfy the requirements on thrust but also that they be capable of carrying out multimissions into outer space with, ideally, the least amount of maintenance and component replacements between missions. This has imposed a greater emphasis on preventing failures and prolonging the fatigue life expectancy of the SSME during operation. Furthermore, specific knowledge relating to the frequency characteristics of the pressure fluctuations in rocket engine systems is not well understood. It is important to understand the functional relationships between pressure fluctuations in these systems in terms of the geometrical and performance characteristics because of their effect on hardware lifetime.

The SSME has been subjected to extensive hot firing and flow tests. During these tests, system failures and malfunctions have occurred from time to time. These failures range from wear on component bearings and flow-induced failures to explosions due to intense pressure fluctuations and accompanying dynamic stresses in the pumps, valves, and/or propellant lines. The large volume of data on pressure fluctuations, strain, and acceleration obtained from these tests provide a good basis for studies on the flow-induced vibrations taking place in the SSME.

The sheer volume of test data available makes it absolutely essential that a computerized data base management system be employed to perform any meaningful analysis. Wyle Laboratories is engaged in the development of such a data base management software. This paper provides an overview of the work being performed on the software development and discusses the method of approach suggested for analyzing the data.

SSME Data Base Management

The SSME data base management and analysis system developed at Wyle Laboratories has been designed for use on an Interdata 8/32 computer system and other computer systems utilizing similar file name structures.[1]

Data Filing System

The data base management system is designed to accommodate two types of data: time histories and power spectral density or other frequency domain data. The time history (rms pressure or thrust) data files consist of a pair of values in each record, the first value being the number of seconds into the test and the second being the magnitude. The file may consist of any number of data points (records), but most of the routines presently have an upper limit of 500 points.

Each PSD file consists of a single real number in each record. These are the spectral magnitudes. Since the PSD test data is recorded using constant bandwidth filters, the frequencies corresponding to each PSD value is automatically calculated by the program when the frequency range is entered. A considerable saving in computer disc storage space has been realized by this method because of the large amount of PSD data involved.

Data entry into files can be performed through a digitizing tablet. When the volume of data is large, however, use of a digitizing tablet can be very time consuming. Hence the data in this case is directly read from a digital tape using the dedicated magnetic tape drive units. File names are automatically assigned by the software based on test numbers, engine numbers, channel descriptors, etc., to uniquely identify each data set. Figure 1 provides an insight into the operational sequences and capabilities of the SSME data base management and analysis software.

Figure 2 shows a plot of power versus time, and figure 3 shows the rms pressure variations with time for the low pressure fuel pump. These plots were generated using the data retrieval system described above. Figures 4 shows a PSD plot generated from the digitized data for the high pressure oxidizer pump. The input to the computer to generate these plots are the file names for the time history plots and the test

number, time slice, and maximum frequency for the PSD plots.

Statistical parameters, such as mean, standard deviation, rms values, third and fourth central moments, and minimum and maximum values and range, can be calculated and plotted with ease for any number of data sets selected based on engine, HPFP, and HPOP numbers; thrust level; and channel descriptors. The abscissa and the ordinate can have any combination of linear and logarithmic scales, and any statistical parameter can be plotted against any other or a frequency scale. The uniqueness of the set of PSDs selected for statistical analysis can be controlled by specifying or leaving out, in any combination, the engine, HPOP, and HPFP numbers.

A detailed description of all the capabilities of the data base management and analysis software is presented in reference 1.

General Character of Turbomachinery Noise

Background

The jet noise theory gained importance with increasing use of jet engines in aircraft. It is well known that the acoustic power generated by a jet exhaust is strongly dependent on the exit velocity (Lighthill's 8th power law[2]). However, during approach configuration of the aircraft, when the jet noise level is generally low, the engine compressor noise was observed to be a major noise source. The study of compressor noise, especially the multistage axial flow types, has acquired great importance during the last 25 years.

Noise Sources in Axial Flow Compressors

In general, the noise spectra of different types of compressors exhibit similar gross characteristics. In this paper, we limit our attention to axial flow compressors because it is believed that their noise characteristics should be more relevant to the noise

generation mechanism of the axial flow pumps on the SSME.

The general form of an acoustic spectrum of an axial flow compressor would indicate broadband noise extending over a wide range of frequencies. Superimposed on this are a number of discrete peaks which represent the blade passage frequencies and their harmonics.

The broadband noise component has been attributed to various mechanisms.[3] At least three that merit consideration are

1. The random force fluctuations due to a moving flow on the surface. These fluctuations can be interpreted to act as acoustic sources located within the turbulent boundary layer.

2. The shedding of vortices from the trailing edges of a body immersed in a moving fluid. This vortex shedding imparts to the body fluctuating lift forces that are periodic at low Reynolds number flows and become random as the Reynolds number increases. (It is believed that the periodic component of the shedding does not disappear completely.) This random excitation force of vortex shedding is also considered to be instrumental in the broadband characteristic of the acoustic spectrum.

3. A turbulent flow at the inlet to the compressor sets up fluctuating forces due to flow incidence on the surface. The perturbation velocity component in a turbulent flow is random in nature, and hence the acoustic energy due to the fluctuating forces can be expected to have a broadband nature.

Another significant broadband component in the case of the SSME would be the structurally transmitted secondary vibrations due to the close proximity of the components, all having highly turbulent flow taking place within them. A challenging task in the analysis of SSME component vibration is the development of a prediction scheme that will provide quantitative

estimates of the acoustic energy due to flow-induced and structurally transmitted vibrations in a frequency spectrum at various locations on the SSME.

The pure tones, which are a result of the spinning modes of the pressure field, exhibit decaying trends downstream of the rotor when the impeller tip Mach number is below critical, as in the case of a subsonic rotor. For a rotor-stator combination, however, a number of spinning modes are generated, many of which spin at Mach numbers above the critical tip Mach numbers, and these modes exhibit strong propagation characteristics.[4]

SSME Flow-Induced Vibration Characteristics

In this section, preliminary studies conducted on pressure spectra obtained from SSME firing tests are discussed. Fluid flow variables, which may be of lesser importance in conventional axial flow compressors, but which may be of greater significance in the SSME flow environment, are highlighted.

The initial efforts in the data analysis have been directed toward obtaining nondimensional PSD as a function of Strouhal number. The power spectral densities, which have dimensions of mean square pressure per hertz, were normalized as follows:

$$S_p = Q_p(f)/\rho_f^2 V^3 D \qquad (1)$$

where S_p is nondimensional PSD, $Q_p(f)$ is the measured PSD, ρ_f is the local fluid density, V is the mean velocity, and D is a characteristic length. Strouhal number is defined as

$$S_t = fD/V \qquad (2)$$

where f is the frequency.

Figures 5 and 6 show nondimensionalized PSD plotted as a function of Strouhal number for the high pressure oxidizer pump discharge pressure. Similar analyses have been carried out to estimate the

pressure fluctuations due to turbulent flow of water at the wall of a duct. Figure 7, reproduced from reference 5, shows the pressure fluctuation dependence on Strouhal number. Figures 5, 6, and 7 indicate similar trends in the variation of nondimensionalized PSD for increasing Strouhal number. The scaling applied to the SSME data is based on the assumption that the pressure fluctuations are flow induced. This assumption is somewhat oversimplified as it does not address the various other noise-generating mechanisms present during turbomachinery operation.

Dimensional analysis studies of blower noise[6] provide a functional relationship,

$$E = F(D, N, Q, f, \rho, \mu, C) \tag{3}$$

where E is the acoustic power/hertz; D, the blower diameter; N, the rpm; Q, the flow rate; f, the frequency; ρ, the fluid density; μ, the viscosity; and C, the acoustic speed. A similar analysis on the SSME turbopump would require accounting for the acoustic propagation characteristic of the structure.

Another important factor that should be considered is the fluid properties. The fluid flow conditions in the SSME undergo extreme variations in temperature and pressure. The mechanism of propagation of disturbances in fluids at extremely low temperatures and very high pressures requires further studies for proper mathematical modeling. Since the fluid flow conditions experience wide variations in the SSME, it is reasonable to expect variations in local propagation velocities and hence local Mach numbers.

The studies on the flow dynamic environment of the SSME being carried out at Wyle Laboratories are directed toward applying the existing theories on noise generation mechanism in turbomachinery after modifications to reflect the flow environment present in the SSME. An approach that would involve developing an empirical formulation rather than a very rigorous and purely analytical solutions seems to indicate

promise. The basic idea involves modeling the noise generation mechanism due to turbulent flow and the pressure pulses generated by the spinning modes of the pressure field due to rotor-stator interactions separately and combining the two components to obtain the total frequency spectrum. That an empirical approach, though somewhat complicated, is not totally impossible has been amply demonstrated through similar studies performed by Van Niekerk[7] and others.

Acknowledgements

The author wishes to acknowledge the help provided by Mr. T. Coffin in directing the research efforts, Dr. G. T. Meares, the chief architect of the data base software development. This project was sponsored by the NASA, Marshall Space Flight Center under contract No. NAS8-34343.

References

1. Meares, G., T. Coffin, and J. Reinhardt. Flow Dynamic Environments in High Performance Rocket Engines." Wyle Laboratories Research Staff Technical Memorandum TM 84-01, Jan. 1984.

2. Lighthill, M. J. On Sound Generated Aerodynamically, II, Turbulence as a Source of Sound. Proc. Roy. Soc., A222, 1954, pp. 1-32.

3. Sharland, I. J. Sources of Noise in Axial Flow Fans. J. Sound Vib., V. 1, No. 3, July 1964.

4. Tyler, J. M., and T. G. Sofrin. Axial Flow Compressor Noise Studies. Soc. Auto. Engr., Aeronautical meeting, preprint 345D, 1961.

5. Clinch, J. M. Prediction and Measurement of the Vibrations Induced in Thin-Walled Pipes by the Passage of Internal Turbulent Water Flow. J. Sound Vib., V. 12, No. 4, 1970, pp. 429-451.

6. Maling, G. C. Jr. Dimensional Analysis of Blower Noise. J. Acoust. Soc. Am., V. 35, No. 10, Oct. 1963, pp. 1556-1564.

7. Van Niekerk, C. G. Noise Generation in Axial Flow Fans. J. Sound Vib., V. 3, No. 1, Jan. 1966.

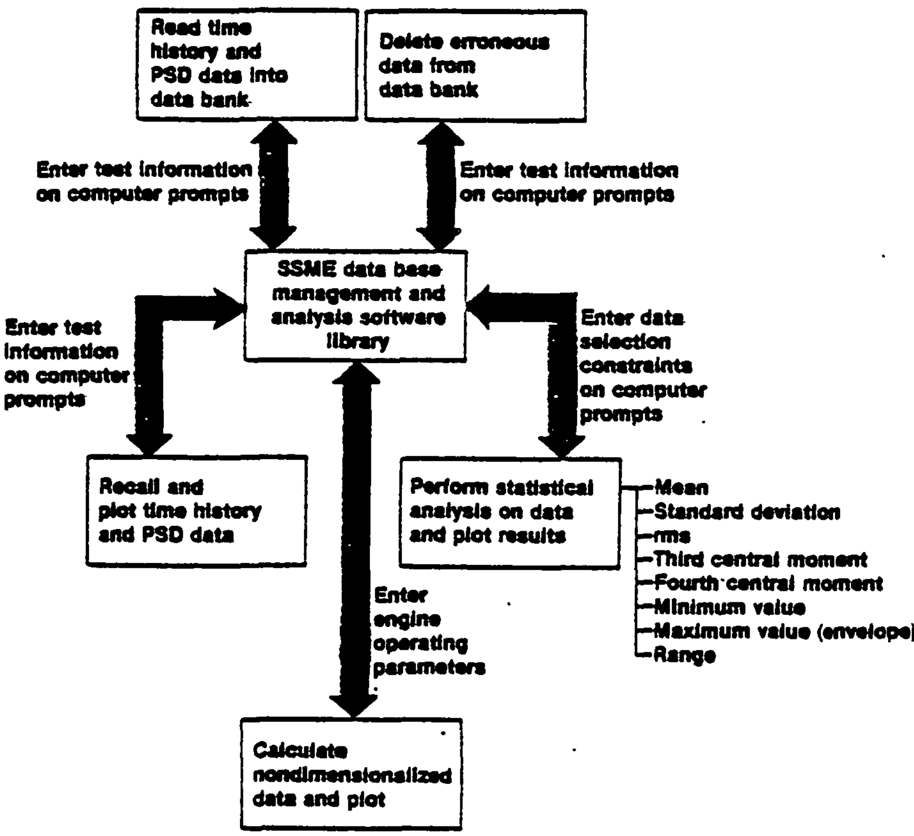


Figure 1. SSME data base management and analysis software operational sequence and capabilities

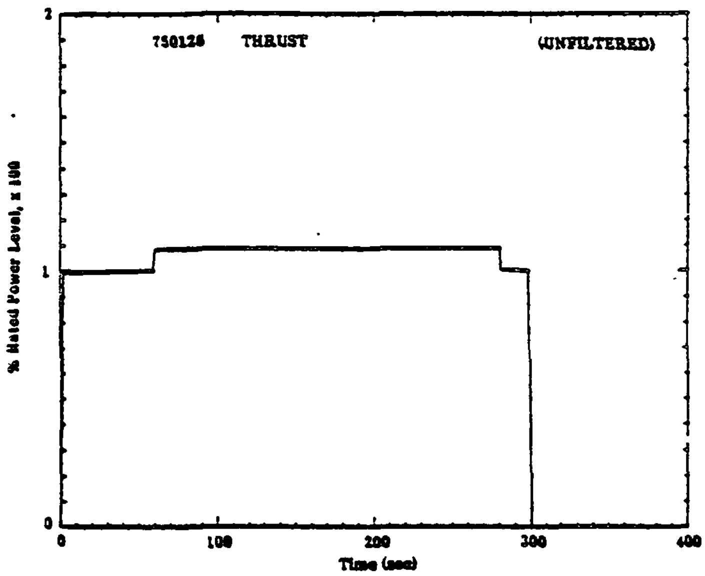


Figure 2. SSME power level variations as a function of time

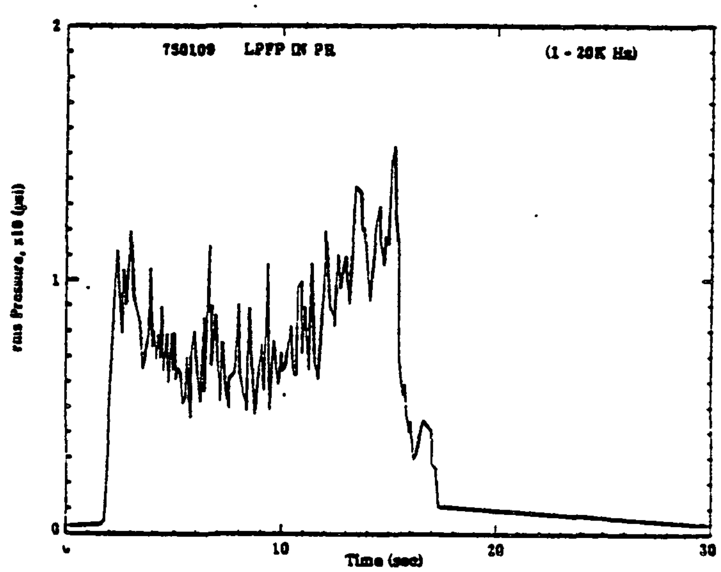


Figure 3. rms pressure time history at the inlet of the low pressure fuel pump

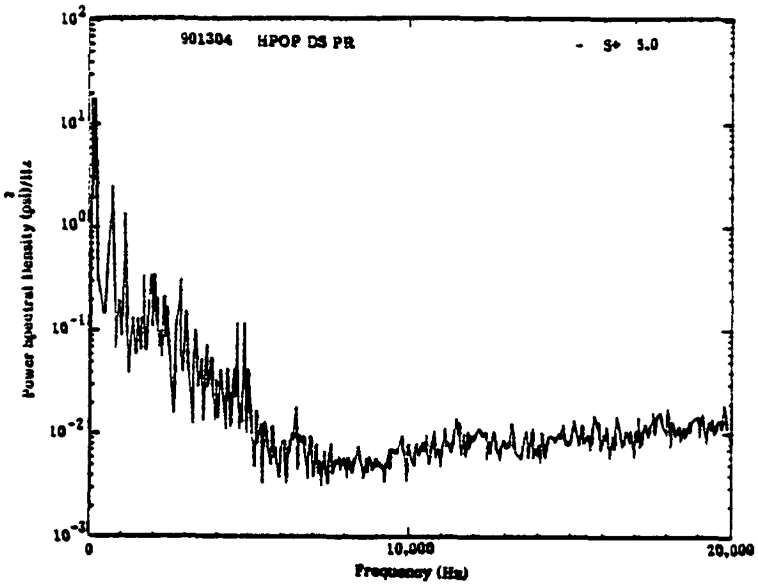


Figure 4. Power spectral density of the high pressure oxidizer pump discharge pressure 5 seconds after SSME startup

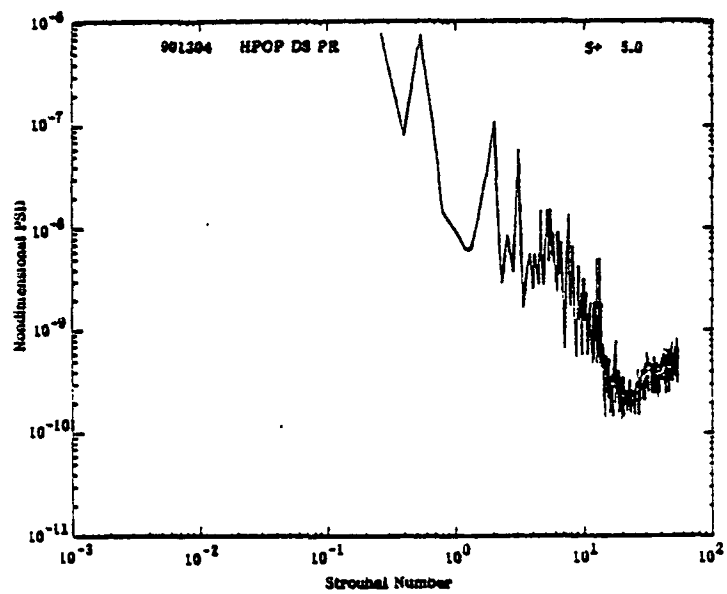


Figure 5. Nondimensionalized frequency spectrum for high pressure oxidizer pump discharge pressure, 5 seconds after SSME startup

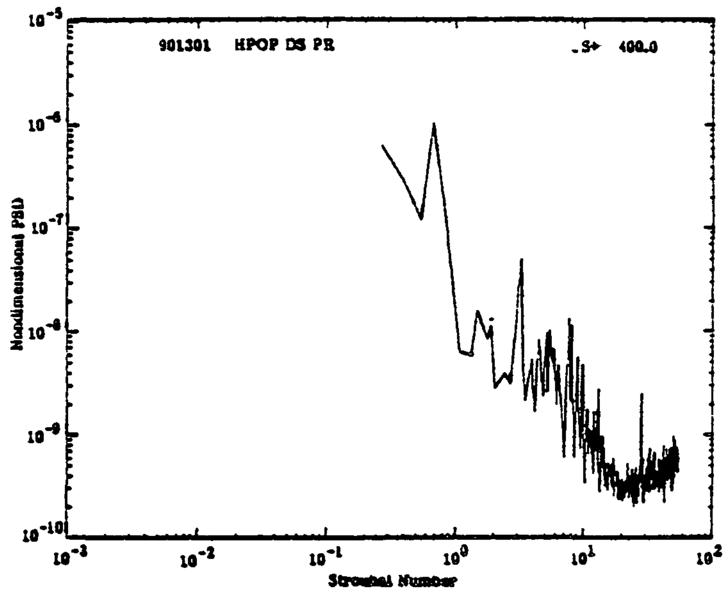


Figure 6. Nondimensionalized frequency spectrum for high pressure oxidizer pump discharge pressure 400 seconds after SSME startup

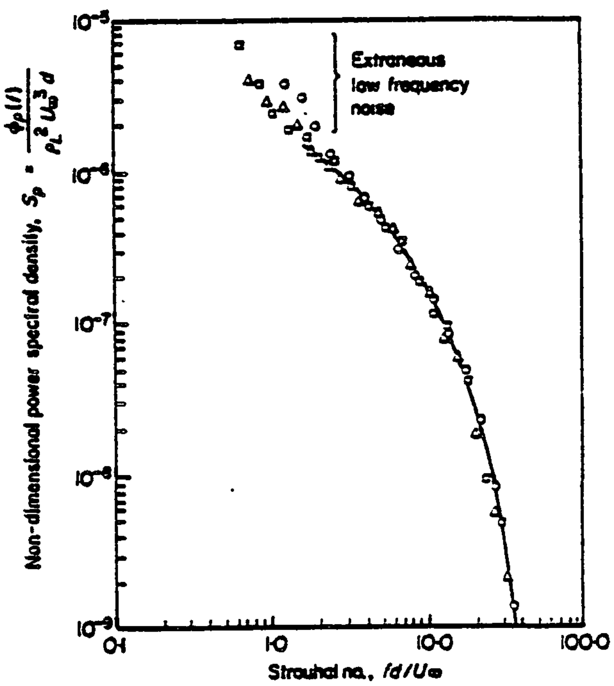


Figure 7. Frequency spectrum of turbulent wall pressure field. (○), 267 in/sec; (△) 450 in/sec; (□) 520 in/sec[5]